

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

3/000 1290

#2

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

JCS78 U.S. PTO
09/938629
08/27/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: 2000年 9月12日

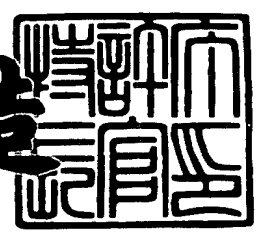
出 願 番 号
Application Number: 特願2000-275983

出 願 人
Applicant(s): 株式会社日立製作所
日立北海セミコンダクタ株式会社

2001年 4月 6日

特許庁長官
Commissioner,
Patent Office

及 川 耕 造



出証番号 出証特2001-3026861

【書類名】 特許願

【整理番号】 H00012901

【提出日】 平成12年 9月12日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 11/10

【発明者】

【住所又は居所】 東京都小平市上水本町五丁目 2 0 番 1 号 株式会社日立製作所 半導体グループ内

【氏名】 矢田 直樹

【発明者】

【住所又は居所】 北海道亀田郡七飯町字中島 1 4 5 番地 日立北海セミコンダクタ株式会社内

【氏名】 石川 栄一

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社日立製作所

【特許出願人】

【識別番号】 000233594

【氏名又は名称】 日立北海セミコンダクタ株式会社

【代理人】

【識別番号】 100089071

【弁理士】

【氏名又は名称】 玉村 静世

【電話番号】 047-361-8861

【手数料の表示】

【予納台帳番号】 011040

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】	図面	1
【物件名】	要約書	1
【プルーフの要否】	要	

【書類名】 明細書

【発明の名称】 データ処理システム及びデータ処理方法

【特許請求の範囲】

【請求項 1】 書き換え可能な不揮発性メモリと、中央処理装置とを有し、前記中央処理装置は、所定の処理を実行し、前記不揮発性メモリの指定された一部の記憶領域の書き換え保証回数を当該不揮発性メモリのその他の記憶領域の書き換え保証回数よりも向上させる処理を行なうものであることを特徴とするデータ処理システム。

【請求項 2】 書き換え可能な不揮発性メモリと、中央処理装置とを有し、前記中央処理装置は、所定の処理を実行して、前記不揮発性メモリの指定された一部の記憶領域に対するライトデータに誤り訂正情報を生成して付加し、前記指定された一部の記憶領域からのリードデータに対し誤り訂正情報による誤り判定と誤り訂正が可能であることを特徴とするデータ処理システム。

【請求項 3】 1 個の半導体チップに前記不揮発性メモリ及び中央処理装置が形成されたシングルチップのマイクロコンピュータであることを特徴とする請求項 1 又は 2 記載のデータ処理システム。

【請求項 4】 前記不揮発性メモリ及び中央処理装置が夫々別々の半導体チップに形成されたマルチチップ形態であることを特徴とする請求項 1 又は 2 記載のデータ処理システム。

【請求項 5】 前記所定の処理は、前記不揮発性メモリの指定された一部の記憶領域への書き込みデータに対して誤り訂正情報を生成する誤り訂正情報生成プログラムと、前記指定された一部の記憶領域から読み出された誤り訂正情報付加データに対するエラー判定とエラーの訂正を行なうエラー訂正プログラムであることを特徴とする請求項 1 又は 2 記載のデータ処理システム。

【請求項 6】 前記データを n ビットとし、 n ビットのデータに対する誤り訂正情報を m ビットとするとき、 m ビットの相互に異なる 2 進数を $m + n$ 列に並べた行列テーブルの記憶領域を有し、前記行列テーブルは前記誤り訂正情報生成プログラム及びエラー訂正プログラムによって参照されるものであることを特徴とする請求項 5 記載のデータ処理装置。

【請求項 7】 前記中央処理装置によってアクセス可能なマスク ROM を有し、前記マスク ROM は、前記誤り訂正情報生成プログラム及びエラー訂正プログラムを保有するものであることを特徴とする請求項 5 記載のデータ処理システム。

【請求項 8】 前記不揮発性メモリのその他の記憶領域は、前記誤り訂正情報生成プログラム及びエラー訂正プログラムの格納領域を有して成るものであることを特徴とする請求項 5 記載のデータ処理システム。

【請求項 9】 前記不揮発性メモリのその他の記憶領域は消去動作が禁止された消去禁止領域と消去及び書き込みが許容された書換え許容領域とを有し、前記前記誤り訂正情報生成プログラム及びエラー訂正プログラムの格納領域は前記消去禁止領域に割当てられて成るものであることを特徴とする請求項 8 記載のデータ処理システム。

【請求項 1 0】 前記誤り訂正情報生成プログラムは、誤り訂正情報を生成した後、生成した誤り訂正情報とそれに対応するデータとを規定のフォーマットに従い誤り訂正情報付加データとして前記指定された一部の記憶領域に格納するものであり、前記エラー訂正プログラムは前記規定のフォーマットに従って誤り訂正情報付加データを認識するものであることを特徴とする請求項 7 又は 9 記載のデータ処理システム。

【請求項 1 1】 前記不揮発性メモリのその他の記憶領域は消去動作が禁止された消去禁止領域と消去及び書き込みが許容された書換え許容領域とを有し、前記誤り訂正情報生成プログラム及びエラー訂正プログラムの格納領域は前記書換え許容領域に割り当てられて成るものであることを特徴とする請求項 8 記載のデータ処理システム。

【請求項 1 2】 前記不揮発性メモリから前記誤り訂正情報生成プログラム及びエラー訂正プログラムが転送される RAM を有し、前記中央処理装置は前記 RAM に転送された前記誤り訂正情報生成プログラム及びエラー訂正プログラムを実行するものであることを特徴とする請求項 5 記載のデータ処理システム。

【請求項 1 3】 前記中央処理装置はリセットの指示に応答して前記不揮発性メモリから前記 RAM に前記誤り訂正情報生成プログラム及びエラー訂正プロ

グラムを転送するものであることを特徴とする請求項 1 2 記載のデータ処理システム。

【請求項 1 4】 前記中央処理装置によってアクセス可能な R A M を有し、前記中央処理装置はリセットの指示に応答して、前記不揮発性メモリの一部の記憶領域から順次誤り訂正情報付加データを読み出し、読み出した誤り訂正情報付加データに対し、前記エラー訂正プログラムの実行により前記エラー判定とエラーの訂正を行い、前記エラー判定とエラーの訂正処理を経たデータを前記 R A M に初期的にストアするものであることを特徴する請求項 5 又は 7 記載のデータ処理システム。

【請求項 1 5】 前記中央処理装置は、前記エラー訂正プログラムの実行による前記エラー判定処理において訂正不能なエラーの発生を示す情報を外部から認識可能に保持する手段を有して成るものであることを特徴とする請求項 5 又は 7 記載のデータ処理システム。

【請求項 1 6】 演算制御装置のアドレス空間に書換え保証回数の低い第 1 記憶領域と、書換え保証回数の高い第 2 記憶領域とを有し、前記第 1 記憶領域は、前記第 2 記憶領域への書き込みデータに対して E C C コードを生成する E C C コード生成プログラムと、前記第 2 記憶領域から読み出された E C C コード付加データに対するエラー判定とエラーの訂正を行なうエラー訂正プログラムとを有し、前記演算制御装置は、前記第 2 記憶領域にデータを格納するとき、前記 E C C コード生成プログラムを実行するものであることを特徴とするデータ処理システム。

【請求項 1 7】 前記演算制御装置は、前記第 2 記憶領域からデータを読み出すとき、前記エラー訂正プログラムを実行するものであることを特徴とする請求項 1 6 記載のデータ処理システム。

【請求項 1 8】 前記演算制御装置は、所定の動作モードに応答して前記エラー訂正プログラムを実行して前記第 2 記憶領域のデータを予め R A M に順次転送可能であることを特徴とする請求項 1 6 記載のデータ処理システム。

【請求項 1 9】 前記第 1 記憶領域はマスク R O M であり、前記第 2 記憶領域は電氣的に書き換え可能なフラッシュメモリであることを特徴とする請求項 1

6 記載のデータ処理システム。

【請求項 2 0】 前記第 1 記憶領域及び第 2 記憶領域は電氣的に書換え可能なフラッシュメモリであり、前記フラッシュメモリは当該フラッシュメモリに対する書き込み消去プログラムを保有し、更に、前記フラッシュメモリから前記書き込み消去プログラムが転送される R A M を有し、前記演算制御装置は特定の動作モードに応答して R A M 上の前記書き込み消去プログラムを実行可能であることを特徴とする請求項 1 6 記載のデータ処理システム。

【請求項 2 1】 n ビットのデータに対して E C C コードを m ビットとするとき、 m ビットの相互に異なる 2 進数を $m + n$ 列に並べた行列テーブルを用いるデータ処理方法であって、E C C コードの生成では、データの論理値 “1” のビット位置に対応する前記行列テーブルの列の値を行方向のビット毎に排他的論理和を採り、これによって得られた m ビットの値を E C C コードとし、データに E C C コードを付加して $m + n$ ビットの符号語を生成することを特徴とするデータ処理方法。

【請求項 2 2】 前記符号語の論理値 “1” のビット位置に対応する前記行列テーブルの列の値を行方向のビット毎に排他的論理和を採り、これによって得られた m ビットの値が全ビット論理値 “0” のときはエラー無と判定して前記符号語の n ビットのデータを正規データとし、前記排他的論理和によって得られた m ビットの値が 1 ビットでも論理値 “1” のときはエラー有りと判定して、前記行列テーブルの列から、前記排他的論理和によって得られた m ビットの 2 進数に一致する列を検索し、検索された列に対応される位置の符号語のビットを論理値反転して訂正し、訂正された符号語の n ビットのデータを正規データとする、ことを特徴とする請求項 2 1 記載のデータ処理方法。

【請求項 2 3】 前記中央処理装置は、前記エラー訂正プログラムの実行中にエラー訂正可能なデータを検出した場合、その検出結果に対応する情報を保持する記憶回路を有することを特徴とする請求項 5 又は 7 記載のデータ処理システム。

【請求項 2 4】 上記検出結果に対応する情報は、ワーニング情報として利用されることを特徴とする請求項 2 3 記載のデータ処理システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、不揮発性メモリに対する書換え保証回数を改善する技術に関し、例えば電氣的に書き換え可能なフラッシュメモリを内蔵するマイクロコンピュータに適用して有効な技術に関する。

【0002】

【従来の技術】

フラッシュメモリなどの電氣的に書き換え可能な不揮発性メモリ（以下単にフラッシュメモリとも称する）はメモリセルにプログラムされる閾値電圧の相違に応じて情報を記憶する。閾値電圧の相違はフローティングゲートが保有する電子又は正孔の量の違いによって得られる。斯くフラッシュメモリセルの電子又は正孔の保持性能は書換え回数の増加と共に劣化する。したがって、記憶情報の信頼性という観点から、通常、フラッシュメモリの使用には有限の書換え保証回数が考慮される。

【0003】

例えば100回程度の書換え保証回数に対してそれを上回る書換えを可能にするには、100回の書換え回数毎に、フラッシュメモリの記憶エリアを順次切換え制御して対処することが可能であるが、そのためには、実使用容量に対して数10～数1000倍にも及ぶ大きな記憶容量が必要になってしまう。

【0004】

フラッシュメモリの書換え保証回数を改善するには、デバイスの手法として、ゲート酸化膜を厚くして電子又は正孔の保持性能を上げることが可能である。また、回路的な手法としてECC（エラー・チェック・アンド・コレクト）回路を採用することが可能である。特開平11-296392号公報には1チップマイクロコンピュータに内蔵のEEPROMにECCを適用した技術が示される。

【0005】

【発明が解決しようとする課題】

しかしながら、前記ゲート酸化膜を厚くすることによる書換え保証回数の改善

には限界が有り、しかも書き込み時間の大幅な増大が予想される。

【 0 0 0 6 】

また、フラッシュメモリにハードウェアでECC機能を付加する場合には、ECCコードの生成回路と、ECCコード付加データに対する誤り判定及び訂正の回路が設けられる。しかしながら、その場合には、ECC用のハードウェアによってチップ面積が増える。その上、必然的に全てのデータに対してECCコードを付加してデータ記憶を行なうことになるので、それに応じてフラッシュメモリ自体の記憶容量も大きくされることになる。特開平11-296392号公報記載の技術はECCコードの生成をCPUのファームROMを用いてソフトウェアで行なうことにより、その分の専用ハードウェアは削減されている。一方、ECCコード付加データに対する誤り判定及び訂正を行なうハードウェア回路は依然設けられている。しかも、全ての記憶データに対してECCコードを付加してデータ記憶が行なわれるようになっている。

【 0 0 0 7 】

特に本発明者は、組込み機器制御用途のマイクロコンピュータ等におけるフラッシュメモリの利用形態を考慮した。例えば、フラッシュメモリには、比較的书換え回数の少ないデータとして例えば、回路特性調整用のトリミングデータ、参照用のテーブルデータ、プログラムデータが記憶される。さらにその他に、機器の状態に応じたパラメータデータのような頻繁に書き換えることが必要なデータもフラッシュメモリに記憶されることがある。一つのフラッシュメモリにそれらデータを混在させようとするとき、従来の技術では全てのデータに対してECCコードが付加されるため、記憶エリアの利用効率が著しく悪化してしまう。頻繁に書き換えるべきデータ量が少ない場合には特に記憶エリアの利用効率の低下が顕著となるであろう。不揮発性メモリの記憶容量が増大する傾向にあるとき、本発明者は一部の記憶領域だけをECCの対象にするという観点の有用性を見出した。

【 0 0 0 8 】

また、フラッシュメモリのメモリセル特性はプロセスばらつきの影響を受けるので、同じ書換え回数でも読み出しエラーを生じ易いメモリセルと生じ難いメモ

リセルがある。本発明者は、そのメモリセル特性の個体差に着目して、データビット数に対するECCコードのビット数をフラッシュメモリの特性に応じて決定できるようにすることの有用性を見出した。要するに、一定の書換え保証回数を得るのに、メモリセルのデバイス特性に合ったECC方式を選択できるようにして、データに対するECCコードのオーバーヘッドを小さくして記憶領域の利用効率を最大限とすることの有用性を見出した。本発明者の検討によれば、このECC方式の選択という融通性は、ECCコードの生成又は誤り検出及び訂正をハードウェアで行なう場合には容易に実現し難いということが明らかにされた。

【 0 0 0 9 】

本発明の目的は、誤り訂正情報例えばECCコードによる記憶領域の利用の無駄を省いて記憶情報の信頼性を向上させることが可能なデータ処理システムを提供することにある。

【 0 0 1 0 】

本発明の別の目的は、ECCコードによる記憶領域の利用の無駄を省いて記憶情報の書換え保証回数を向上させることが可能なデータ処理システムを提供することにある。

【 0 0 1 1 】

本発明の更に別の目的は、デバイスの特性に合ったECC方式を選択できるようにして、データに対するECCコードのオーバーヘッドを小さくして記憶領域の利用効率を最大限とすることが可能なデータ処理システムを提供することにある。

【 0 0 1 2 】

本発明の更に別の目的は、ECCコードが付加されたデータに対する誤り判定及び訂正の処理によるデータリード動作の遅延を最小限にすることが可能なデータ処理システムを提供することにある。

【 0 0 1 3 】

本発明のその他の目的は、ECCコードの生成を効率的に行なうことができるデータ処理方法を提供することにある。

【 0 0 1 4 】

本発明のその他の目的は、ECCコードが付加されたデータの誤り判定を効率的に行なうことができるデータ処理方法を提供することにある。

【0015】

本発明のその他の目的は、電氣的に消去及びプログラム可能な不揮発性メモリのアドレス空間の一部のアドレス領域の書き換え回数をソフトウェア的処理によって向上することが可能なデータ処理システムないしデータ処理方法を提供することにある。

【0016】

本発明の前記並びにその他の目的と新規な特徴は本明細書の記述及び添付図面から明らかになるであろう。

【0017】

【課題を解決するための手段】

本願において開示される発明のうち代表的なものの概要を簡単に説明すれば下記の通りである。

【0018】

〔1〕データ処理システムは、書き換え可能な不揮発性メモリ（フラッシュメモリ）と、中央処理装置（CPU）とを有し、前記中央処理装置は、所定の処理（プログラム）を実行し、前記不揮発性メモリのアドレス空間内の指定された一部の記憶領域（20Ba）の書き換え保証回数を当該不揮発性メモリのアドレス空間内のその他の記憶領域の書き換え保証回数よりも向上させる処理が可能である。前記所定のプログラムの実行による処理は、前記指定された一部の記憶領域に対するライトデータに誤り訂正情報（ECCコード）を生成して付加する処理であり、前記不揮発性メモリの指定された一部の記憶領域からのリードデータに対しECCコードによる誤り判定と誤り訂正を行なう処理である。前記一部の記憶領域の指定は例えばユーザプログラムによって行なわれる。

【0019】

上記より、不揮発性メモリの指定された一部の記憶領域に格納されたデータに対するアクセスだけを対象として、ECCコードの付加や誤り訂正を行なって書き換え保証回数を向上させる。前記指定された一部の記憶領域には書き換えの頻繁な

パラメータデータが格納され、他の記憶領域には書換え頻度の低いプログラムデータなどが格納される。したがって、当該他の記憶領域に格納されるデータには E C C コードが付加されない。この構成は、記憶領域に拘わらず全てのライトデータに対して区別なく E C C コードを付加する構成に比べて、実質的に無用の E C C コードによる記憶領域の無駄な利用を省きながら記憶情報の信頼性を向上、若しくは記憶情報の書換え保証回数の向上を実現する。

【 0 0 2 0 】

ライトデータに対する E C C コードの付加、並びに E C C コードによる誤り判定及び訂正の処理を中央処理装置によるプログラム実行で実現するから、データビット数に対する E C C コードのビット数の割合を定義する E C C 方式はプログラムの記述内容で選択できる。そのため、メモリセルのデバイス特性に合った E C C 方式の選択が容易であり、データに対する E C C コードのオーバーヘッドを小さくできる。その結果記憶領域の利用効率を最大限とすることが可能になる。要するに、ソフトウェアによりエラー訂正効率を容易に変えることができるから、デバイスの能力に合ったエラー訂正方式を容易に選択することができる。観点を変えれば、これは E C C コードのビット数から無駄を排除し、記憶エリアの有効利用を保証する。

【 0 0 2 1 】

前記データ処理システムは、1 個の半導体チップに前記不揮発性メモリ及び中央処理装置が形成されたシングルチップのマイクロコンピュータとして実現することが可能である。一方、前記データ処理システムは、前記不揮発性メモリ及び中央処理装置が夫々別々の半導体チップに形成されたマルチチップ形態で実現してもよい。

【 0 0 2 2 】

中央処理装置が実行する前記所定のプログラムは、例えば、前記不揮発性メモリの指定された一部の記憶領域への書き込みデータに対して E C C コードを生成する E C C コード生成プログラムと、前記不揮発性メモリの指定された一部の記憶領域から読み出された E C C コード付加データに対するエラー判定とエラーの訂正を行なうエラー訂正プログラムである。

【 0 0 2 3 】

ECCによる符号化や誤り訂正の公知の手法では例えばハミングコードの検査行列を用い、誤り訂正ではリードデータとハミングコードの行列演算を積和演算などを用いて行なうことができる。ここでは、データを n ビットとし、 n ビットのデータに対するECCコードを m ビットとすると、 m ビットの相互に異なる2進数を $m+n$ 列に並べた行列テーブルを記憶領域に形成し、前記ECCコード生成プログラム及びエラー訂正プログラムの処理にその行列テーブルを参照させる。

【 0 0 2 4 】

例えば、ECCコードの生成では、データの論理値“1”のビット位置に対応する前記行列テーブルの列の値を行方向のビット毎に排他的論理和を採り、これによって得られた m ビットの値をECCコードとし、データにECCコードを付加して $m+n$ ビットの符号語を生成する。エラー判定及びエラー訂正では、前記符号語の論理値“1”のビット位置に対応する前記行列テーブルの列の値を行方向のビット毎に排他的論理和を採り、これによって得られた m ビットの値が全ビット論理値“0”のときはエラー無と判定して前記符号語の n ビットのデータを正規データとし、前記排他的論理和によって得られた m ビットの値が1ビットでも論理値“1”のときはエラー有りと判定して、前記行列テーブルの列から、前記排他的論理和によって得られた m ビットの2進数に一致する列を検索し、検索された列に対応される位置の符号語のビットを論理値反転して訂正し、訂正された符号語の n ビットのデータを正規データとする。このエラー判定では積和演算と除算演算のような処理を要せず、積和演算器及び除算器を備えずともソフトウェアによるECC処理を能率的に行なうことが可能になる。

【 0 0 2 5 】

前記前記ECCコード生成プログラム及びエラー訂正プログラムは前記中央処理装置によってアクセス可能なマスクROMに保持させてよい。それらプログラムの保持に前記不揮発性メモリを利用する場合には、その他の記憶領域に保持せればよい。パラメータに比べてプログラムの書換え頻度は少ないから相対的に書換え保証回数の小さなその他の記憶領域に保持せれば充分である。要するに

、劣悪な使用条件を除けば、書換え頻度の低い情報にECCコードを付加する実益は殆ど無いということである。

【 0 0 2 6 】

前記不揮発性メモリのその他の記憶領域に、消去動作が禁止された消去禁止領域（20A）と消去及び書き込みが許容された書換え許容領域（20B）とを割当てるとき、消去禁止領域に前記ECCコード生成プログラム及びエラー訂正プログラムを格納してよい。その場合には、一旦書き込まれたプログラムの書換えは原則不可能であるから、望ましくは、半導体集積回路化されたマイクロコンピュータのようなデータ処理装置の製造メーカーが書き込むのがよい。そうであるなら、データとECCコードとのフォーマットは予め固定的に決定された方が望ましく、前記ECCコード生成プログラムは、ECCコードを生成した後、生成したECCコードとそれに対応するデータとを規定のフォーマットに従いECCコード付加データとして前記不揮発性メモリの一部の記憶領域に格納し、エラー訂正プログラムは前記規定のフォーマットに従ってECCコード付加データを認識すればよい。

【 0 0 2 7 】

また、前記ECCコード生成プログラム及びエラー訂正プログラムを前記書換え許容領域に保持させてもよい。この場合は、ユーザによるそれらプログラムの書き込みを想定する。したがって、データとECCコードとのフォーマットについてはユーザが作成したプログラムに依存した任意フォーマットであってもよい。製造メーカーがそれらプログラムを提供しようとする場合にもユーザの使い勝手を考慮すれば、C言語のような高級言語で記載されたソースプログラムで与えるのがよい。データとECCコードとを別々の配列として把握することも可能になる。

【 0 0 2 8 】

ECC処理速度を向上させる一つの手段として、ランダムアクセスメモリ（RAM）のアクセスサイクルが前記不揮発性メモリのアクセスサイクルより速い場合、前記不揮発性メモリから前記ECCコード生成プログラム及びエラー訂正プログラムをRAM転送し、前記中央処理装置には前記RAMに転送された前記E

ＣＣコード生成プログラム及びエラー訂正プログラムを実行させればよい。このとき、前記中央処理装置はリセットの指示に応答して前記不揮発性メモリから前記ＲＡＭに前記ＥＣＣコード生成プログラム及びエラー訂正プログラムを転送するとよい。

【 0 0 2 9 】

ＥＣＣ処理速度を向上させる第２の手段として、前記中央処理装置によってアクセス可能なＲＡＭを有し、前記中央処理装置はリセットの指示に応答して、前記不揮発性メモリの一部の記憶領域から順次ＥＣＣコード付加データを読み出し、読み出したＥＣＣコード付加データに対し、前記エラー訂正プログラムの実行により前記エラー判定とエラーの訂正を行い、前記エラー判定とエラーの訂正処理を経たデータを前記ＲＡＭに初期的にストアするとよい。その後、ＣＰＵはＲＡＭから必要なデータをリードすればよく、リード動作に際してその都度エラー判定を行なうことを要しない。尚、ＣＰＵによる上記エラー訂正プログラムの実行する方法としては、上記不揮発性メモリから直接上記エラー訂正プログラムを読み込んで実行する方法又はＲＡＭに転送された上記エラー訂正プログラムを読み込んで実行する方法のいずれかを採用することが可能である。

【 0 0 3 0 】

前記エラー訂正プログラムの実行による前記エラー判定処理において訂正不能なエラーが発生したき、誤動作防止の観点より、前記中央処理装置は、前記エラー訂正プログラムの実行による前記エラー判定処理において訂正不能なエラーの発生を示す情報を外部から認識可能にレジスタ（ＲＥＲ）、或いはメモリのフラグ領域（３０）に保持させるとよい。ユーザプログラムを介してその領域を所定インターバル毎に参照すれば、訂正不能なエラー発生を認識でき、例えば、訂正不能なエラーが発生したデータブロックの書換え等を実行して対処することが可能になる。

【 0 0 3 1 】

また、前記中央処理装置は前記エラー訂正プログラムの実行によりエラー訂正可能なデータを検出した場合にも、それを示す情報を外部から認識可能に汎用レジスタやメモリのフラグ領域に保持させてよい。そのような情報は上記と同じく

ワーニング情報として利用すればよい。これにより、ユーザシステムは、データが壊れかけている、ということを逸早く認識することができ、壊れかけているデータの書換え（再書込み）を促してデータの信頼性を更に向上させることが可能になる。

【 0 0 3 2 】

〔 2 〕 本発明の別の観点によるデータ処理システムは、演算制御装置のアドレス空間に書換え保証回数の低い第 1 記憶領域（ 2 0 B b ）と、書換え保証回数の高い第 2 記憶領域（ 2 0 B a ）とを有し、前記第 1 記憶領域は、前記第 2 記憶領域への書き込みデータに対して ECC コードを生成する ECC コード生成プログラムと、前記第 2 記憶領域から読み出された ECC コード付加データに対するエラー判定とエラーの訂正を行なうエラー訂正プログラムとを有し、前記演算制御装置は、前記第 2 記憶領域にデータを格納するとき、前記 ECC コード生成プログラムを実行する。

【 0 0 3 3 】

この観点では、第 2 記憶領域に対する書き込み動作に対してだけ ECC コードをソフトウェアで生成するから、上記同様に、実質的に無用の ECC コードによる記憶領域の無駄な利用を省きながら記憶情報の信頼性を向上、若しくは記憶情報の書換え保証回数の向上を実現することができる。更に、ライトデータに対する ECC コードの付加をプログラム実行で実現するから、やはり、不揮発性メモリのメモリセルのデバイスの特性に合った ECC 方式の選択が容易であり、データに対する ECC コードのオーバーヘッドが小さくでき、記憶領域の利用効率を最大限とすることが可能にある。

【 0 0 3 4 】

前記演算制御装置は、前記第 2 記憶領域からデータを読み出すとき、前記エラー訂正プログラムを実行する。

【 0 0 3 5 】

前記演算制御装置は、所定の動作モードに応答して前記エラー訂正プログラムを実行して前記第 2 記憶領域のデータを予め R A M に順次転送可能である。

【 0 0 3 6 】

前記第1記憶領域及び第2記憶領域は例えば電氣的に書換え可能なフラッシュメモリである。ECCコード生成プログラム及びエラー訂正プログラムの書換えを考慮すると、書き込み消去プログラムを前記フラッシュメモリに予め格納し、所定の動作モードに応答してこれをRAMに内部転送し、RAM上の書き込み消去プログラムを実行してECCコード生成プログラム及びエラー訂正プログラムの書換えを行なうようにしてよい。或いは、所定の動作モードに応答して外部からRAMに書き込み消去プログラムを転送し、RAM上の書き込み消去プログラムを実行してECCコード生成プログラム及びエラー訂正プログラムの書換えを行なうようにしてよい。

【0037】

〔3〕ECCコードを用いた符号語を生成するデータ処理方法では、 n ビットのデータに対してECCコードを m ビットとすると、 m ビットの相互に異なる2進数を $m+n$ 列に並べた行列テーブルを利用し、ECCコードの生成では、データの論理値“1”のビット位置に対応する前記行列テーブルの列の値を行方向のビット毎に排他的論理和を採り、これによって得られた m ビットの値をECCコードとし、データにECCコードを付加して $m+n$ ビットの符号語を生成する。

【0038】

上記符号語に対するエラー判定を行なうデータ処理方法では、前記符号語の論理値“1”のビット位置に対応する前記行列テーブルの列の値を行方向のビット毎に排他的論理和を採り、これによって得られた m ビットの値が全ビット論理値“0”のときはエラー無と判定して前記符号語の n ビットのデータを正規データとし、前記排他的論理和によって得られた m ビットの値が1ビットでも論理値“1”のときはエラー有りと判定して、前記行列テーブルの列から、前記排他的論理和によって得られた m ビットの2進数に一致する列を検索し、検索された列に対応される位置の符号語のビットを論理値反転して訂正し、訂正された符号語の n ビットのデータを正規データとする。このエラー判定では積和演算ないし除算演算のような処理を要せず、算術論理演算器やシフタなどの演算器及び除算器を有し積和演算器を備えていないCPUを用いてもソフトウェアによるECC処理

を能率的に行なうことが可能になる。

【0039】

【発明の実施の形態】

《マイクロコンピュータ》

図1には本発明の一例に係るシングルチップのマイクロコンピュータが示される。同図に示されるマイクロコンピュータ1は、特に制限されないが、単結晶シリコンのような1個の半導体基板（半導体チップ）にCMOS集積回路製造技術により形成される。

【0040】

マイクロコンピュータ1は、演算制御装置としての中央処理装置（CPU）2、RAM3、バスステートコントローラ（BSC）4、電氣的に書き換え可能な不揮発性メモリとしてのフラッシュメモリ5、フラッシュコントロールモジュール6、及びその他の内蔵回路を総称するその他モジュール7を有する。その他モジュール7としてマスクROM8、割り込みコントローラ（INTC）9、タイマ（TMR）10、入出力ポート（I/O）11及びシリアルインタフェースコントローラ（SCI）12等を有する。それら回路モジュールはバスIAB、IDB、PAB、PDB、CONTを介してインタフェースされる。

【0041】

前記バスIAB、IDBは情報伝送速度の比較的速い内部アドレスバス、内部データバスである。前記バスPAB、PDBは情報伝送速度が比較的遅い周辺アドレスバス、周辺データバスである。バスCONTはバスアクセス制御信号やタイミング制御信号等を伝達する制御信号線を総称する。内部バスIDB、IABと周辺バスPDB、PABとの動作速度の相違若しくはアクセス対象に固有のアクセス形態の相違に対して前記BSC4がアクセス動作タイミング等を最適制御すると共に、前記BSC4はアクセスアドレスに応じたチップ選択若しくはモジュール選択制御等も行なう。

【0042】

前記CPU2は、特に制限されないが、マスクROM8又はRAM3から命令をフェッチし、フェッチした命令を解読して実行する。RAM3はCPU2のワ

ーク領域若しくはデータ又はプログラムの一時記憶領域とされる。前記マスクROM 8はプログラム又はデータテーブルなどの記憶領域とされる。割り込みコントローラ 10はマイクロコンピュータ 1の外部から与えられる割込要求又はマイクロコンピュータ 1内部の状態に応じて内蔵回路モジュールから発生される割込要求を受け、割り込み優先度及び割り込みマスク等に従って割込要求の受付を調停する。割込要求が受け付けられると、CPU 2に割り込み信号IRQが与えられ、割り込みベクタによって割り込み要因がCPU 2に与えられる。CPU 2は割り込みベクタによって指示されるプログラムに処理を分岐する。I/O 11は外部アドレスバス及び外部データバスへの接続、SCI 12の外部インタフェース、TMR 10の外部イベント信号入力等に用いられる。

【 0 0 4 3 】

前記CPU 2の具体例は図2に示される。CPU 2は、特に制限されないが、シフタSFT及び算術論理演算器ALU等の演算器と、32ビットの汎用レジスタR0～R31、プログラムカウンタPC、コンディションコードレジスタCCR及びテンポラリレジスタTR等のレジスタ群、そしてリードデータバッファRDB、ライトデータバッファWDB及びアドレスバッファABなどのバッファ回路を実行部に有する。命令制御部は命令レジスタIR、命令デコーダIDEC、命令シーケンスロジックINTLを有する。

【 0 0 4 4 】

前記プログラムカウンタPCは次に実行すべき命令アドレスを保有し、その命令アドレスがアドレスバッファABから内部アドレスバスIABに出力されると、RAM 3等の対応アドレスからリードされた命令が内部データバスIDBを介して命令レジスタIRにフェッチされる。命令デコーダIDECは命令レジスタIRの命令を解読して、CPU 2内部の制御信号を生成して、前記実行部による演算処理を制御する。命令シーケンスロジックINTLは割り込み信号IRQ等に応答して命令実行順序を変更する制御を行なう。

【 0 0 4 5 】

図1においてフラッシュメモリ5は、メモリセルアレイ20、Xデコーダ・ドライバ(XDE・DV) 21、センスアンプアレイ(SAA) 22、Yスイッチ

アレイ (Y S W) 2 3、Yデコーダ (Y D E) 2 4、入出力回路 (I F B) 2 5、電源回路 (V G N) 2 6、及びタイミングジェネレータ (T G N) 2 7を有する。メモリセルアレイ 2 0はマトリクス配置されたフラッシュメモリセル (図示せず) を有する。フラッシュメモリセルは、特に制限されないが、半導体基板若しくはウェル領域にソース、ドレインを有し、チャネルの上方に夫々絶縁膜を介してフローティングゲート及びコントロールゲートが形成されたスタック構造を有し、ソースをソース線に、ドレインをビット線に、コントロールゲートをワード線に接続して構成される。

【 0 0 4 6 】

フラッシュメモリセルは閾値電圧がプログラム可能にされ、プログラムされた閾値電圧に応じて情報を保持する。例えば、1個のフラッシュメモリセルが1ビットの情報を保持する場合に、相対的に高い閾値電圧状態を書き込み状態、相対的に低い閾値電圧状態を消去状態と称する。書き込み状態を得る為の書き込み動作は、特に制限されないが、コントロールゲートに10V、ドレインに例えば5V、ソースおよび基板に例えば0Vを印加して、ドレイン・ソース間に電流を流し、これによってホットエレクトロン注入が起こり、フローティングゲートに電子が蓄積され、メモリセルの閾値電圧が高くなる。前記消去状態を得る為の消去動作は、特に制限されないが、コントロールゲートに10V、ソース及び基板に例えば-10Vを印加し、さらにドレインを例えば開放 (フローティング) にして、フローティングゲートに蓄積された電子を基板に放出させ、これによってメモリセルの閾値電圧が低くなる。

【 0 0 4 7 】

前記入出力回路 2 5はバス I A B, I D B, P A B, P D B, C O N Tとの間でアドレス、制御信号及びコマンドを入力すると共にデータの入出力を行なう。入出力回路 2 5に入力されたアドレス信号はX D E C・D V 2 1及びY D E 2 4に入力されて夫々デコードされる。X D E C・D V 2 1はそのデコード結果に従ってワード線を選択する。Y D E 2 4はそのデコード結果に従ってY S W 2 3を介してビット線を選択する。ワード線選択及びビット線選択によってフラッシュメモリセルが選択される。読み出し動作では、前記選択されたフラッシュメモリ

セルの読み出しデータは、S A A 2 2 にて検出され、入出力回路 2 5 を経てバス P D B または I D B に出力される。書き込み動作では、バス P D B 又は I D B から入出力回路 2 5 に与えられる書き込みデータが入出力回路 2 5 内の書き込みラッチ回路にラッチされ、ワード線選択されたメモリセルに対し、ラッチデータに従って書き込み・書き込み阻止が制御される。書き込み処理の前には予めブロック単位でフラッシュメモリセルに対する消去が行なわれる。

【 0 0 4 8 】

前記電源回路 2 6 はクランプ回路やチャージポンプ回路などを有し、フラッシュメモリの書き込み・消去・読み出しなどの動作で使用する様々な電圧を供給する。前記タイミングジェネレータ 2 7 は、制御バス C O N T を介して供給されるストローク信号及びデータバス P D B, I D B を介して入力されるコマンドに基づいてフラッシュメモリの内部タイミング信号を生成する。

【 0 0 4 9 】

図 1 において前記フラッシュコントロールモジュール 6 は、フラッシュメモリ 5 に対する書き込み及び消去のためのシーケンス制御並びに E C C 処理に利用される回路ブロックである。このフラッシュコントロールモジュール 6 は、C P U 2 によってアクセス可能な夫々 3 2 ビットの、書き込み／消去制御レジスタ F L M C R、消去ブロック指定レジスタ E B R、データレジスタ F M P D R 0, F M P A R 0、及びリザルトレジスタ F P F R 等の制御レジスタを備え、更に、フラッシュメモリに対する書き込み及び消去のシーケンス動作を制御するシーケンス制御回路 2 9 を有する。

【 0 0 5 0 】

前記書き込み／消去制御レジスタ F L M C R はフラッシュメモリ 5 の動作モードを制御するレジスタであり、書き込みの有効／無効を指示するライトイネーブルビット W E、消去動作を指示するイレーズビット E、消去ベリファイ動作を指示するイレーズベリファイビット E V、書き込み動作を指示するプログラムビット P、書き込みベリファイ動作を指示するプログラムベリファイビット P V、フラッシュメモリの書き込み動作中にエラーの発生したことを示す書き込みエラービット P E R、消去動作中にエラーの発生したことを示す消去エラービット E E

R、フラッシュメモリのリード動作時にエラーの発生したことを示すリードエラービット R E R 等を有する。消去ブロック指定レジスタ E B R はフラッシュメモリセルアレイ 2 0 の消去エリアをブロック毎に設定するレジスタであり、ブロック毎にイレーズブロックビット E B 0 ~ E B 9 を有する。データレジスタ F M P D R 及びリザルトレジスタ F P F R 等は後述の E C C 処理で利用されるレジスタである。尚、レジスタ F M P D R 及び F P F R は、汎用レジスタ（R 0 - R 3 1）内のレジスタを利用しても良い。

【 0 0 5 1 】

《一部のエリアに対するソフトウェア E C C 処理》

次にマイクロコンピュータ 1 の E C C 機能について説明する。フラッシュメモリ 5 のメモリセルアレイ 2 0 は、特に制限されないが、図 3 の（A）に例示されるようにブート領域 2 0 A とユーザ領域 2 0 B に大別される。特に制限されないが、ブート領域 2 0 A はマイクロコンピュータ 1 のユーザによる自由な書換えが禁止される領域であり、ユーザ領域 2 0 B はユーザによる自由な書換えが許容される領域である。要するに、前記消去ブロック指定レジスタ E B R のイレーズブロックビット E B 0 ~ E B 9 の設定によって消去可能なエリアはユーザ領域 2 0 B に限られる。ブート領域に格納されたプログラムはユーザが所定の動作モードを設定することにより実行可能である。尚、フラッシュメモリに対する書換えシーケンスの具体例についてここでは詳述しないが、書き込みデータにしたがった書き込み動作は先に消去動作が完了されることを条件とするものであり、消去ブロックによる消去が指示されなければ当然書き込みデータに従った書き込み動作も行なわれないようになっている。そのような書き込み・消去制御は前記シーケンス制御回路 2 9 が行なう。

【 0 0 5 2 】

マイクロコンピュータ 1 の E C C 機能は、図 3 の（B）に例示されるように、前記ユーザ領域 2 0 B に指定された一部の領域 2 0 B a のデータに向けられている。即ち、E C C 機能は、C P U 2 が、所定のプログラムを実行することにより、ユーザ領域 2 0 B の一部の記憶領域（第 1 領域） 2 0 B a の書換え保証回数を当該ユーザ領域 2 0 B のその他の記憶領域（第 2 領域） 2 0 B b の書換え保証回

数よりも向上させる機能である。この所定のプログラム実行による E C C 機能は、前記指定された一部の記憶領域 2 0 B a に対するライトデータに E C C コードを生成して付加する処理（E C C コード生成処理）と、前記指定された一部の記憶領域 2 0 B a からのリードデータに対し E C C コードによるエラー判定とエラー訂正を行なう処理（E C C エラー判定訂正処理）とによって実現される。前者の処理は C P U 2 が E C C 生成プログラム 2 1 を実行することにより行なわれ、後者の処理は C P U 2 がエラー訂正プログラム 2 2 を実行することにより行なわれる。

【 0 0 5 3 】

前記領域 2 0 B a には頻繁に書き換えられる制御用パラメータなどのデータの記憶に利用される。頻繁に書き換えられる制御用パラメータとは、たとえばマイクロコンピュータ 1 が自動車のエンジン制御に用いられる場合、エンジン停止時の各ピストンの相対位置、エンジン停止直前までの燃費情報等の情報とされる。頻繁に書き換えられるデータに E C C 処理を施せば、仮に、書き換え回数の増加によるフラッシュメモリセルの特性劣化に起因して読み出しデータに誤りを生じても、E C C コードによる誤り訂正能力の範囲で、その誤りを訂正することができる。換言すれば、E C C 処理により実質的に書き換え保証回数を改善若しくは向上させることができる。

【 0 0 5 4 】

前記領域 2 0 B b は、図 3 の（B）では、頻繁に書き換えられることのない情報として、テーブルデータなどの固定データ、前記 E C C 生成プログラム 2 1、エラー訂正プログラム 2 2 及び他のユーザプログラム等の記憶領域として利用される。前記 E C C 生成プログラム 2 1 及びエラー訂正プログラム 2 2 は、図 3 の（C）に例示されるようにブート領域 2 0 A に格納し、或いは図 3 の（D）に例示されるようにマスク R O M 8 に格納してもよい。

【 0 0 5 5 】

上記より、ユーザ領域 2 0 B の一部の記憶領域 2 0 B a に対するアクセスだけを対象として、E C C コードの付加や誤り訂正を行なって書き換え保証回数を向上させるから、前記一部の記憶領域 2 0 B a には書き換えの頻繁なパラメータデータ

等を格納すればよい。他の記憶領域 2 0 B b には書換え頻度の低いプログラムデータなどを格納しても、当該他の記憶領域には E C C コードが付加されないから、記憶領域に拘わらず全てのライトデータに対して区別なく E C C コードを付加する構成に比べて、実質的に無用の E C C コードによる記憶領域の無駄な利用を省きながらフラッシュメモリセルの記憶情報の信頼性向上、換言すれば記憶情報の書換え保証回数の向上を実現することができる。

【 0 0 5 6 】

ライトデータに対する E C C コードの付加、並びに E C C コードによる誤り判定及び訂正の処理を C P U 2 によるプログラム実行で実現するから、ユーザデータビット数に対する E C C コードのビット数の割合を定義する E C C 方式は E C C コード生成プログラム及びエラー訂正プログラムの記述内容で選択でき、フラッシュメモリのメモリセルのデバイス特性に合った E C C 方式の選択が容易である。例えば、図 4 に例示されるようにユーザデータビット数に対する E C C コードのビット数が減るほど訂正能力は低くなるが E C C コードによって費やされるメモリセルの割合（オーバヘッド）は少なくなる。したがって、E C C 処理のソフトウェアによりエラー訂正効率を容易に変えることができるから、マイクロコンピュータのデバイス能力に合ったエラー訂正方式を容易に選択することができ、これは E C C コードのビット数から無駄を排除することを意味し、フラッシュメモリの記憶エリアの有効利用を保証する。要するに、ユーザデータに対する E C C コードのオーバヘッドを小さくしてフラッシュメモリの利用効率を最大限とすることが可能になる。

【 0 0 5 7 】

《 E C C 処理領域の規定 》

前記ユーザ領域 2 0 B の一部の記憶領域 2 0 B a を E C C 処理対象とするとき、その記憶領域 2 0 B a を規定するのは、例えば記憶領域 2 0 B b に格納されたユーザプログラムである。前記記憶領域 2 0 B a にパラメータを格納するとき、例えばユーザプログラムは、パラメータデータのソースアドレスと領域 2 0 B a のストア先アドレスを指定し、C P U の処理を E C C コード生成プログラムにジャンプさせる。E C C コード生成プログラムが実行されることにより、ソースア

ドレスのパラメータデータに対してECCコードが生成され、パラメータデータとしてのユーザデータにECCコードを付加したECC付きデータが前記ストア先アドレスに格納される。パラメータデータをリードするときは、ユーザプログラムは領域20Baのソースアドレスと任意のディスティネーションアドレスを指定し、CPUの処理をエラー判定プログラムにジャンプさる。エラー判定プログラムが実行されることにより、ソースアドレスのECC付きデータがリードされ、これに対する誤り判定が行なわれ、必要な訂正が行なわれてパラメータデータとしてのユーザデータがディスティネーションアドレスにセットされる。ECC処理を伴った上記パラメータデータの格納及びパラメータデータのリードにおいて、ユーザプログラムとECCコード生成プログラム及びエラー訂正プログラムとの間で行なわれるべきアクセスアドレス及びリードデータの受け渡しは、CPU2の汎用レジスタ或いはRAM3の領域を介して行なわれる。この詳細は、ソフトウェアECC処理の具体例で説明する。

《ユーザデータとECCコードとの対応》

上記処理において、ユーザデータとECCコードとの対応はユーザプログラム、ECCコード生成プログラム及びエラー訂正プログラムにおいて統一的に把握されていることが望ましい。例えば、図5に例示されるように、ECCコード生成対象とされるユーザデータの配列DA1に対し、ユーザデータと対応するECCコードを一つのレコードとするように一つの配列データDA2としてフォーマット化し、或いは、ユーザデータの配列DA1に対してECCコードを別のデータ配列DA3として規定してよい。後者の場合は当然、ユーザデータの配列DA1とECCコードの配列DA3は先頭アドレスなどによって配列相互の対応付けが必要であることは言うまでもない。

【0058】

図6にはユーザデータとECCコードとを一つの配列データ中に対応付けて領域20Baに書き込むときの処理手順が例示される。まず、書き込み対象とされるユーザデータの配列が指定されてECCコード付きデータの書き込み処理が指示されると、ユーザデータの読み込みと展開が行なわれる(S1)。即ち、(B)に例示されるように、ユーザデータの配列DA1がワークメモリに読み込まれ

、そこで、読み込まれたユーザデータは規定のデータフォーマットにしたがって、(B)の配列EXTで例示されるように、ユーザデータの隣にECCコード領域を有するレコード形式に拡張される。次いで、(A)に例示されるようにユーザデータに対してECCコードを生成し、これを対応レコードのECCコード領域にストアして、レコード配列DA2が形成される。このレコード配列DA2のデータに対して前記フラッシュメモリの領域20Baへの書き込みが行なわれる(S3～S7)。書き込み処理は、書き込みパルスの印加(S4)、書き込みデータのペリファイ(S5)、ペリファイ結果に基づく書き込み終了判定(S6)を行ない、所望の書き込み状態に到達していなければ再書き込みデータを演算してステップS4からの処理を繰り返し、規定回数繰り返してもステップS6の終了判定が満足できなければ書き込み異常終了とされ、ステップS6で規定の書き込み状態に到達すれば書き込み正常終了とされる。書き込みパルス印加の前には書き込みエリアに対する消去が終了されているものとする。

【0059】

図7にはユーザデータとECCコードとを別の配列データとして対応付けて領域20Baに書き込むときの処理手順が例示される。まず、書き込み対象とされるユーザデータの配列が指定されてECCコード付きデータの書き込み処理が指示されると、ユーザデータの読み込みが行なわれる(S11)。即ち、(B)に例示されるように、ユーザデータの配列DA1がワークメモリに読み込まれる。次いで、(A)に例示されるようにユーザデータに対してECCコードを生成し、これを別のデータ配列DA3としてストアする(S12)。双方のデータ配列DA1、DA3に対して前記フラッシュメモリの領域20Baへの書き込みが行なわれる(S13～S17)。書き込み処理は前記ステップS4～S7の処置と同様である。

【0060】

図8には図6で説明したレコード配列を有するECCコード付きデータをリードするときの処理手順が例示される。まず、リード対象とされるECCコード付きデータのレコード先頭アドレスが設定され(S21)、これに対応するECCコード付きデータがフラッシュメモリの領域20Baからワークメモリに読み込

まれる（S 2 2）。読み込まれたデータに対してエラー判定が行なわれる（S 2 3）。このとき、図 6 で説明したようにエラー判定の為に読み込まれた配列データは一定のフォーマット、即ち、ユーザデータと ECC コードの所定ビット数単位のペアを単一レコードとする配列のフォーマットを有しているから、エラー訂正プログラムは、その規定のフォーマットであることを前提に、各レコードからユーザデータと ECC コードを参照して誤りの判定を行なうことができる。訂正可能な誤りに対してはエラー訂正が実行され（S 2 4）、必要な訂正が実行されたリードデータが RAM 3 などの所定のエリアにストアされる（S 2 5）。

【 0 0 6 1 】

図 9 には図 7 で説明したユーザデータと ECC コードが別々の配列データとされる ECC コード付きデータをリードするときの処理手順が例示される。先ず、リード対象とされる ECC コードの配列データの先頭アドレスとユーザデータの先頭アドレスが設定され（S 3 1）、これに対応する ECC コードとユーザデータがフラッシュメモリの領域 2 0 B a からワークメモリに読み込まれ（S 3 2）、読み込まれたデータに対してエラー判定が行なわれる（S 3 3）。エラー判定において、対応するユーザデータと ECC コードを参照する場合にも夫々の配列の構造を指定する情報がユーザプログラムを介して与えられなければならない。要するに、図 7 で説明した ECC コード用配列データ DA 3 の生成に際して、当該配列データ DA 3 の先頭アドレス及び構造はユーザプログラムを介して与えられているから、その配列を利用する場合にも同じようにユーザプログラムから必要なアドレス情報及び配列の構造情報が与えられなければならない。訂正可能な誤りに対してはエラー訂正が実行され（S 3 4）、必要な訂正が実行されたリードデータが RAM 3 などの所定のエリアにストアされる（S 3 5）。

【 0 0 6 2 】

図 3 の（B）で説明したように、前記 ECC コード生成プログラム及びエラー訂正プログラムを前記書換え許容領域としてのユーザ領域 2 0 B に保持させれば、それらプログラムをユーザが開発して書き込むことができるという自由度を得る。したがって、ユーザデータと ECC コードとのフォーマットについてはユー

ザ作成のプログラムに依存した任意フォーマットにする方がユーザにとって都合がよい場合もある。このとき、マイクロコンピュータ 1 のメーカーがそれら ECC コード生成プログラム及びエラー訂正プログラムを提供しようとする場合にもユーザの使い勝手を考慮すれば、C 言語のような高級言語で記載されたソースプログラムで与えるのがよい。そのような任意フォーマットを考慮したとき、ECC コードの付加及び ECC コード付きデータを用いたエラー訂正処理方式には、例えば図 7 及び図 9 で説明した手順を採用してよい。

【 0 0 6 3 】

一方、図 3 の (C) で説明したように、消去禁止領域としてのブート領域 2 0 A に前記 ECC コード生成プログラム及びエラー訂正プログラムを格納することを想定する。この場合には、ブート領域 2 0 A に一旦書き込まれたプログラムの書換えは原則不可能であるから、望ましくは、それらプログラムはマイクロコンピュータ 1 のメーカーが書き込むのがよい。そうであるなら、ユーザデータと ECC コードとのフォーマットに関しても、ユーザに対する自由度の保証という観点は低く、逆にユーザの負担軽減という観点より、固定フォーマットを採用する方が得策であると考えられる。前記 ECC コード生成プログラムは、ECC コードを生成した後、生成した ECC コードとそれに対応するデータとを規定のフォーマットに従い ECC コード付加データとして前記不揮発性メモリの一部の記憶領域に格納し、エラー訂正プログラムは前記規定のフォーマットに従って ECC コード付加データを認識すればよい。そのような固定フォーマットを考慮したとき、ECC コードの付加及び ECC コード付きデータを用いたエラー訂正処理方式として、例えば図 6 及び図 8 で説明した手順を採用すればよい。前記 ECC コード生成プログラム及びエラー訂正プログラムを図 3 の (D) で説明したマスク ROM 8 に格納する場合も同様に考えてよい。

【 0 0 6 4 】

《オンボードプログラムモード》

オンボードプログラムモードについて説明する。この動作モードは、図 3 の (B) で説明したように前記 ECC コード生成プログラム及びエラー訂正プログラム等をユーザ領域 2 0 B に保持させる時に必要な動作モードの一例である。

【 0 0 6 5 】

先ず、ブートモードによるオンボードプログラムの手順を説明する。パーソナルコンピュータ又はE P R O Mライタ等のホスト装置に書き込み制御プログラムと、前記E C Cコード生成プログラム及びエラー訂正プログラムを用意し、I / O 1 1の所定のポートに接続する。所定の外部端子を規定の状態にしてマイクロコンピュータ1をブートモードに遷移させる。ブートモードに遷移すると、マイクロコンピュータ1はブート領域のブートプログラムを実行し、S C I 1 2による通信を可能とし、ブート領域からR A M 3に、フラッシュメモリ5のユーザ領域2 0 Bに対する消去プログラムと、S C I 1 2を介する通信制御プログラムをロードする。ついで、ロードされた消去プログラムが実行されてユーザ領域2 0 Bが全面消去され、前記通信制御プログラムによってホストから書き込み制御プログラムがR A M 3にロードされる。その後、書き込み制御プログラムが実行され、ホストが保有する前記E C Cコード生成プログラム及びエラー訂正プログラム等がユーザ領域2 0 Bに書き込まれる。

【 0 0 6 6 】

次に、ユーザプログラムモードによるオンボードプログラムの手順を説明する。パーソナルコンピュータのようなホスト装置に書き込み・消去制御プログラムと、前記E C Cコード生成プログラム及びエラー訂正プログラムを用意し、また、ユーザ領域2 0 Bにはホスト装置と間の転送制御プログラムを予め格納しておく。先ず、C P U 2は割込みに応答し或いはジャン命令を実行することにより前記転送制御プログラムを実行し、ホスト装置から書き込み・消去制御プログラムをR A M 3に転送する。次に、R A M 3上で書き込み・消去制御プログラムを実行し、ユーザ領域2 0 Bの必要なエリアを消去し、そこに、ホストが保有する前記E C Cコード生成プログラム及びエラー訂正プログラムを書き込む。

【 0 0 6 7 】

《E C C処理速度の向上》

図1 0には前記E C Cコード生成プログラム及びエラー訂正プログラムの実行速度を向上させる為の一例が示される。即ち、前記フラッシュメモリ5のブート領域2 0 A（図3の（C））又はユーザ領域2 0 B（図3の（B））から前記E

CCコード生成プログラム及びエラー訂正プログラムをRAM3の所定のアドレス領域へ転送し、前記CPU2には前記RAM3に転送された前記ECCコード生成プログラム及びエラー訂正プログラムを実行させればよい。このとき、前記CPU2はリセットの指示に応答して前記フラッシュメモリ5から前記RAM3への転送制御プログラムを実行すればよい。そのような転送制御プログラムは例えばユーザ領域20Bbにおけるユーザプログラムとして、或いはブート領域20Aのプログラムとして所定のアドレス領域に格納しておけばよい。この方法は、RAM3のアクセスサイクルがフラッシュメモリ5のアクセスサイクルより速い場合において、上記ECCコード生成プログラム及びエラー訂正プログラムのCPUによる実行速度を高速にすることができる。

【0068】

図11にはエラー判定によるオーバーヘッドを見掛け上解消するための一例が示される。即ち、前記CPU2はリセットの指示に応答して、所定のリセット処理ユーザプログラムを実行し、前記フラッシュメモリ5の一部の記憶領域20Baから順次ECCコード付加パラメータデータDATA1を読み出し、読み出したECCコード付加パラメータデータDATA1に対し、前記エラー訂正プログラム22の実行により前記エラー判定とエラーの訂正を行い、前記エラー判定とエラー訂正処理を経たパラメータデータDATA2を前記RAM3に初期的にストアする。その後、CPU2は制御用ユーザプログラムにしたがってRAM3から必要なデータをリードすればよく、リード動作に際してその都度エラー判定を行なうことを要しない。

【0069】

なお、図11の前記エラー訂正プログラム22は、図10で示されるように、フラッシュメモリ5の所定のアドレス領域からRAM3の所定のアドレス領域へ転送された状態を示している。

【0070】

図12は、エラー判定によるオーバーヘッドを見掛け上解消する為の第2の方法の一例が示される。上記図11は前記エラー訂正プログラム22をRAM3のアドレス空間上でCPU2により実行する場合を示したが、図12はRAM3の

リードアクセスサイクルがフラッシュメモリ 5 のリードアクセスサイクルより速くない（短くない）場合に有効な方法が示される。

【 0 0 7 1 】

フラッシュメモリ 5 のアクセスサイクルと RAM 3 のアクセスサイクルとが同一の場合、エラー訂正プログラム 2 2 の CPU 2 による実行速度は、RAM 3 の記憶領域上からエラー訂正プログラムを CPU 2 で実行しても、フラッシュメモリの記憶領域上からエラー訂正プログラムを CPU 2 で実行しても、その実行速度はそれほど変わらないと考えられる。

【 0 0 7 2 】

したがって、図 1 2 に示されるように、フラッシュメモリ 5 の一部の記憶領域上にエラー訂正プログラム 2 2 を格納し、そのプログラム 2 2 を RAM 3 へ転送することなくフラッシュメモリ 5 の上記一部の記憶領域から CPU 2 で実行し、フラッシュメモリ 5 に一部の記憶領域 2 0 B a に格納された ECC コード付加パラメータ DATA 1 1 に対してエラー判定及びエラー訂正を行ない、エラー判定及びエラー訂正処理の施されたパラメータデータ DATA 2 2 を RAM 3 の所定の記憶領域に格納する。上記エラー訂正プログラムの処理は、上記図 1 1 で説明されたように、フラッシュメモリ 5 の一部の領域 2 0 B b に格納されたりセット処理ユーザプログラムが、リセットに応答する CPU 2 によって実行される事によって実行される。

【 0 0 7 3 】

《ソフトウェア ECC 処理の原理的手法》

次にソフトウェア ECC 処理の原理的な手法を説明する。ECC による符号化や誤り訂正の公知の手法では例えばハミングコードの検査行列を用い、誤り訂正ではリードデータとハミングコードの行列演算を積和演算などを用いて行なうことができる。マイクロコンピュータ 1 におけるソフトウェア ECC 処理では、データを n ビットとし、 n ビットのデータに対する ECC コードを m ビットとするとき、 m ビットの相互に異なる 2 進数を $m + n$ 列に並べた行列テーブルを例えば図 3 の (C) に例示されるフラッシュメモリ 5 の記憶領域 2 0 B b に固定データとして形成し、前記 ECC コード生成プログラム及びエラー訂正プログラムの処

理にその行列テーブルを参照させる。

【 0 0 7 4 】

図 1 3 には 1 6 ビットデータに対する 1 ビット訂正のための行列テーブル 4 0 の一例が示される。同図の行列テーブルは便宜上生成行列 4 1 と検査行列 4 2 に分けて図示されている。生成行列 4 1 における行番号 1 ～ 2 1、列番号 1 ～ 1 6 は、1 ～ 1 6 が 1 6 ビットデータに関する番号、1 7 ～ 2 1 が 5 ビットの ECC コードに関する番号であると理解されたい。検査行列 4 2 における行番号 1 ～ 2 1 は生成行列と同意義、列番号 1 ～ 5 は 5 ビットの ECC コードに関する番号であると理解されたい。

【 0 0 7 5 】

ECC コード付加データ、即ち符号語の生成は次のように行なう。例えば 1 6 ビットのデータを $M = m_{15}, m_{14}, \dots, m_1, m_0$ に対して 5 ビットの ECC コード（検査ビット） $P = p_4, p_3, p_2, p_1, p_0$ を生成して 2 1 ビットの符号語を生成する。この符号語を生成するには生成行列 4 1 とデータ M との行列演算を行なえばよい。要するに、図 1 4 のように生成行列 4 1 とデータ M を並べ、データ M のビットが “1” に対応する生成行列 4 1 の行を加算すれば良い。ここで行なう加算は 2 進数の加算であり、各ビット毎の排他的論理和（E x O R）を計算すればよい。この演算手法において、元の 1 6 ビットのデータ M はそのまま残るから、生成行列 4 1 における ECC コード 5 ビット相当部分に関して排他的論理和を演算すればよい。例えば、 $H' 8041 (b' 1000_0000_0100_0001)$ の 1 6 ビットデータに対して ECC コードを生成するには、図 1 5 に例示されるように、1 6 ビットデータ M に対して、前記生成行列 4 1 の内、ECC コード 5 ビット相当部分に関する行列を抜き出した行列 4 1 A を並べ、データの論理値 “1” のビット位置に対応する前記行列テーブルの列の値を行方向のビット毎に排他的論理和を演算して ECC コードを生成すればよい。図 1 6 にはここで行う排他的論理和演算のための演算手段を ECC コード 1 ビット分の構成を代表として例示する。図 1 6 の構成では、2 ビットの排他的論理和演算器 4 3 の出力をディスティネーションレジスタ（D r e g）4 5 が受け、排他的論理和演算器 4 3 の一方の入力には D r e g 4 3 の出力が帰還され、他

方の入力にはソースレジスタ (S r e g) 4 5 の出力が与えられる。前記排他的論理和演算において D r e g 4 3 の初期値は “0” であり、S r e g 4 3 には順次、データの論理値 “1” のビット位置に対応する前記行列テーブルの列の値が行方向のビット毎に入力され、排他的論理和演算器 4 3 は双方の入力に対する排他的論理和演算結果を D r e g 4 4 にラッチし、これを S r e g 4 3 の次の出力との排他的論理和に用いる。最終的に D r e g 4 4 に得られた結果が、E C C コードの対応ビットの値になる。

【0 0 7 6】

尚、図 1 5、図 1 6 の説明より明らかなように、生成行列 4 1 は少なくとも図 1 4 の列番号 1 7 ~ 2 1 の部分だけあればよい。

【0 0 7 7】

図 1 7 には図 1 5 の場合の排他的論理和演算結果が示される。同図の X e - O R 結果が演算された E C C コードである。H' 8 0 4 1 の場合には、M = b' 1 0 0 0 _ 0 0 0 0 _ 0 1 0 0 _ 0 0 0 1 に P = b' 1 1 1 0 1 を付加したデータ C = 1 0 0 0 _ 0 0 0 0 _ 0 1 0 0 _ 0 0 0 1, 1 1 1 0 1 が、2 1 ビットの符号語になる。

【0 0 7 8】

フラッシュメモリから読み出された符号語に対するエラー判定は、前記検査行列 4 2 と 2 1 ビットの符号語との積を計算する。実際に積を計算しようとするれば積和演算による行列演算を行なわなければならない。ここでは、図 1 8 に例示されるように、検査行列 4 2 に対して符号語 C を並べ、前記符号語の論理値 “1” のビット位置に対応する前記検査行列 4 2 の列の値を行方向のビット毎に排他的論理和を採る。この排他的論理和演算は図 1 6 で説明したのと同じ手法で行えばよい。図 1 8 の例はフラッシュメモリ 5 から読み出された符号語に誤りの無い場合を示しており、上記生成された符号語 C = 1 0 0 0 _ 0 0 0 0 _ 0 1 0 0 _ 0 0 0 1, 1 1 1 0 1 と同じ符号語が読み出された状態が示されている。

【0 0 7 9】

図 1 9 には図 1 8 の例において前記排他的論理和による演算結果が示される。図 1 9 の結果の欄に示された 5 ビットの値 R 1 t が全ビット論理値 “0” のとき

はエラー無と判定され、前記符号語Cに含まれる16ビットのデータMが正規データとされる。

【0080】

図20にはフラッシュメモリ5から読み出した符号語C_{er}に1ビットの誤りを含む場合が例示される。図20の結果の欄に示された5ビットの値R_{1t}が1ビットでも論理値“1”のときはエラー有りと判定される。誤りのあるビット位置は、前記検査行列42の列から、前記排他的論理和によって得られた5ビットの2進数R_{1t}に一致する列に対応される位置の符号語C_{er}のビットB_{er}である。この誤りビットB_{er}に対し、その論理値を反転してエラー訂正を行えばよい。訂正された符号語に含まれる16ビットのデータMが、訂正後の正規データとされる。

【0081】

図20の説明より明らかなように訂正可能な1ビットの誤りを生じているとき、エラー判定の演算結果R_{1t}は検査行列42の何れかの列のビットパターンに一致することになる。エラー判定の演算結果R_{1t}にそれ以外のビットパターンが現れたときは2ビット以上の訂正不能な誤りが発生していることになる。図21にはそのような訂正不能を意味する10種類のビットパターンが示される。検査行列42によるこの例では、それら訂正不能を意味する10種類のビットパターンの値は10進数で22以上の値にされるようになっているから、訂正不能の判定も容易である。

【0082】

以上で説明したソフトウェアECC処理では積和演算を要する行列演算を直接行わないから、CPU2若しくはマイクロコンピュータ1は積和演算器を備えなくとも、ソフトウェアによるECC処理を能率的に行なうことが可能になる。また、行列テーブルを用いるから、その都度生成行列や検査行列を生成しなくてもよい。

【0083】

図22には訂正不能エラーに対して例外処理を実行可能にする例が示される。前記エラー訂正プログラムの実行による前記エラー判定処理において訂正不能な

エラーが発生したとき、誤動作防止の観点より、前記CPU 2は、前記エラー訂正プログラムの実行による前記エラー判定処理において訂正不能なエラーの発生を示す情報を外部から認識可能にRAM 3のフラグ領域FLG 30（或いはフラッシュコントロールモジュール6内のレジスタFMLCRのリードエラービットRER）に保持させる。例外処理プログラムのようなユーザプログラムを介してそのフラグ領域FLGを所定インターバル毎に参照させ、訂正不能なエラー発生を認識したとき、訂正不能なエラーが発生したデータブロック或いは全てのデータの書換え、即ち、新たなユーザデータに、ECCコードを生成し、これを付加して、領域20Baの所定エリアに書き込む処理を実行させる。

【0084】

更にデータ破壊による誤動作防止を更に推し進める場合には、前記CPU 2は前記エラー訂正プログラムの実行によりエラー訂正可能なデータを検出した場合にも、それを示す情報を外部から認識可能にRAM 3の所定記憶領域又は所定の汎用レジスタに保持させるとよい。そのような情報はワーニング情報として利用すればよく、例えば所定のユーザプログラムを介してその情報エリアを定期的に参照させればよい。これにより、ユーザシステムは、データが壊れかけている、ということを逸早く認識することができ、壊れかけているデータの書換え（再書込み）を促してデータの信頼性を更に向上させることが可能になる。要するに実際にデータエラーを生ずる事態の発生を未然に防止することができる。

【0085】

《ECCコード生成処理の具体例》

ここでは図17で説明した固定フォーマットのユーザデータを用意してECCコードを生成する処理を具体的に説明する。ECCコードの生成前に例えばユーザプログラムにしたがってユーザデータの読み込みとフォーマット展開が行なわれる。展開されたデータは図23に例示されるように、ユーザデータ16ビットに対して16ビットの拡張ビットを割り当て、合計32ビットが一つのECCデータブロックとして符号語領域になる。前記16ビットの拡張ビットには5ビットのECCコード領域の他に11ビットのワークビット領域が設けられているが、これは、必要なデータブロックをワード境界を単位にアクセスして得られるよ

うにする実用的な観点を考慮したものである。

【 0 0 8 6 】

図 2 4 には一つの ECC データブロックの構成が更に詳細に示される。同図において符号化検査ビットとは ECC コードを意味する。同図において ECC データブロックのビット番号に対応して機能名が割当てられる。機能名 D 0 0 ~ D 1 5 はユーザデータであり、機能名 P 0 0 ~ P 0 4 は ECC コードである。

【 0 0 8 7 】

図 2 5 には行列テーブルのデータ（生成用データ）例が示される。同図に示される生成用データは前記機能名に対応され、夫々テーブル検索アドレス（検索アドレス） $X \sim X + 14$ が割当てられている。図 2 5 の生成用データは、要するに図 1 8 の検査行列 4 2 と実質的に等しく、この検査行列 4 2 の一部が図 1 5 で説明した実質的な生成行列 4 1 A になっている。図 2 5 の機能名 D 0 0 ~ D 1 5 の生成用データは図 1 8 の列番号 1 6 ~ 1 のビット列に対応され、図 2 5 の機能名 P 0 0 ~ P 0 4 の生成用データは図 1 8 の列番号 2 1 ~ 1 7 のビット列に対応されている。図 2 5 のテーブルを便宜上 ECC テーブル（ECC TLB）とも記す。

【 0 0 8 8 】

図 2 6 には ECC コード生成プログラムによる処理手順が例示される。ECC コード生成プログラムとそれに先立って実行されるユーザプログラムとの間のデータ及びアドレスの受け渡しには CPU 2 の汎用レジスタ及びフラッシュコントロールモジュール 6 内のレジスタが利用される。即ち、ユーザプログラムは、フラッシュコントロールモジュール 6 内のレジスタ FMPAR0 に書き込みデータエリアの先頭アドレスをセットする。要するに、図 2 3 に例示されるところの RAM3 に展開されたデータの先頭アドレスがセットされる。汎用レジスタ R0 はユーザ書き込みデータレジスタ、R1 は行列テーブル検索アドレスポインタ、R2 は行列テーブル検索ストップ値、R3 はユーザ書き込みデータ保存アドレスポインタ、R4 はユーザ書き込みデータ保存アドレスストップ値、R5 はビットマスク用データ、R6 は検査ビット生成用変数、R7 は論理値“1”のビットがあるかを検出するための変数、R8 はビットに対応するテーブル値を格納する変数

、として利用される。

【 0 0 8 9 】

図 2 6 において、先ず、レジスタ FMPDR0 の値をレジスタ R 3 にセットして RAM 3 上に展開されたユーザデータの先頭アドレスをストアする (S 4 0)。次に、レジスタ R 3 の値に h' 8 0 を加算して RAM 上に展開されたユーザデータのストップアドレスをレジスタ R 4 にストアする (S 4 1)。そして、レジスタ R 3 の値がレジスタ R 4 の値に到達するまで、以下の処理を行う (S 4 2)。すなわち、レジスタ R 3 の値を利用して先頭 1 6 ビットユーザデータをリードしてレジスタ R 0 にロードする (S 4 3)。レジスタ R 1 には ECC TLB の先頭の検索アドレス X をセットし、レジスタ R 2 にはそのストップアドレス X + h' 1 0 をセットする (S 4 4)。レジスタ R 5 にビットマスクデータ h' 0 0 0 1 をセットし、レジスタ R 6 にデフォルトデータ h' 0 0 0 0 をセットする (S 4 6)。そしてレジスタ R 1 の値がレジスタ R 2 の値に到達するまで次の処理を繰り返す (S 4 7)。要するに、レジスタ R 7 にレジスタ R 5 の値をロードし、レジスタ R 7 とレジスタ R 0 の値に対して論理積をとり、その結果をレジスタ R 7 に返す (S 4 8)。レジスタ R 7 の値が 0 より大きいかを判定する (S 4 9)。ステップ S 4 9 の判定結果が 0 より大きければ、その時のマスクデータ R 5 の論理値 “1” のビット位置と等しいビット位置に R 0 のユーザデータも論理値 “1” のビットも持つことになる。その場合には ECC テーブルの生成用データ (Y) をレジスタ R 8 にストアし (S 5 0)、レジスタ R 6 と R 8 の値に対して前記排他的論理和処理を行って、その値をレジスタ R 6 に返す。ステップ S 4 9 において R 7 の値が 0 であれば、レジスタ R 5 の値を 1 ビット左シフトしてマスクビット位置を次のビット位置とし (S 5 2)、レジスタ R 1 に h' 0 1 を加算して、ECC TLB の検索アドレスを次アドレスに進める (S 5 3)。そして再度ステップ S 4 8 の処理に戻る。ステップ S 4 8 ~ S 5 3 の処理は ECC TLB の検索アドレスがストップアドレスになるまで繰り返され (S 4 7)、ここまでの処理により、図 1 7 で説明したような ECC コード P がレジスタ R 6 に保持される。そして、次にレジスタ R 3 の値を 1 ワード分だけインクリメントし (S 5 4)、レジスタ R 3 の値が指すアドレス (RAM 3 上に展開された図 2 3 のデー

タフォーマットの拡張領域)に、レジスタR6が保有するECCコードをストアする(S55)。そして次のユーザデータに対して同様の処理を行う為にレジスタR3のアドレスを1ワード分インクリメントする。上記ステップS43～S56までの処理をR3<R4になるまで繰り返すことによって、RAM3上に展開された一群のユーザデータに対してECCコードが付加される。この後、ECCコード生成プログラムの実行を終えて、直前のユーザプログラムにリターンする。特に図示はしないが、リターンされたユーザプログラムは、RAM3上に展開されたECCコード付加データを、フラッシュメモリ5の所定の領域20Baに書き込む。

【0090】

《エラー判定処理の具体例》

図27には図26の手順で生成されたECCコード付加データをリードしたときのエラー判定処理の詳細が示される。エラー判定処理プログラムとそれに先立って実行されるユーザプログラムとの間のデータ及びアドレスの受け渡しにはCPU2の汎用レジスタ及びフラッシュコントロールモジュール6内のレジスタが利用される。即ち、ユーザプログラムは、フラッシュコントロールモジュール6内のレジスタFMPAR0にユーザリードアドレスをセットする。要するに、図3の領域20Ba中のリードアドレスがセットされる。汎用レジスタR0はエラー訂正前のリードアドレス、R1は検査ビット保存変数、R2はエラー検出用レジスタ、R3はユーザリードアドレス、R4はECC計算用中間テーブル、R5はECCTLB検索アドレス、R6はECCTLB検索ストップアドレス、R7はビットマスクデータ、R8は論理値“1”検出用変数、R9はECCTLBデータ保存用変数、として利用される。

【0091】

図27において、先ず、レジスタFMPAR0のユーザリードアドレスをレジスタR3にセットし(S60)、このレジスタR3のリードアドレスを利用してユーザデータをレジスタR0にストアする(S61)。更にアドレスを1ワード分インクリメントし(S62)、後続のECCコードのデータをレジスタR1にストアし、且つレジスタR1のデータをレジスタR4にコピーする(S63)。

前記レジスタR4を左に16ビットシフトを行い、そのシフト結果に対してh' 001F0000との論理積を採る。これによってレジスタR4には、その第17ビット目から第21ビットにECCコードが配置され、その他のビットを“0”としたデータを得る(S64)。更に、レジスタR0の値に対してh' 00000FFFとの論理積を採り、その結果をレジスタR0に返し、レジスタR4の値にレジスタR0の値を加えてその演算結果をレジスタR4に返す(S65)。これにより図示の“1”ビット検索テーブルがレジスタR4に得られる。“1”ビット検索テーブルにおける未使用エリアは論理値“0”になっている。次にレジスタR2を論理値“0”に初期化し(S66)、レジスタR5にECCTLBの検索アドレスXをセットし、レジスタR6に検索ストップアドレスX+h' 16をセットする(S67)。レジスタR7にはビットマスクデータh' 00000001をセットする(S68)。そしてR5の値がR6の値になるまで以下の処理を繰り返す。即ち、レジスタR8にレジスタR4の値をセットし、R7とR8の論理積をR8に返し、R8が0より大きいかを判定する(S71)。ステップS71の判定結果が0より大きければ、その時のR7のマスクデータの論理値“1”のビット位置と等しいビット位置にR4の符号化データも論理値“1”のビットも持つことになる。その場合にはECCテーブルECCTLBの生成用データ(Y)をレジスタR9にストアし(S72)、レジスタR2とR9の値に対して前記排他的論理和処理を行って、その値をレジスタR2に返す(S73)。ステップS71においてR8の値が0であれば、レジスタR7の値を1ビット左シフトしてマスクビット位置を次のビット位置とし(S74)、レジスタR5にh' 01を加算して、ECCTLBの検索アドレスを次アドレスに進める(S75)。そして再度ステップS70の処理に戻る。ステップS70～S77の処理はECCTLBの検索アドレスがストップアドレスになるまで繰り返され(S69)、ここまでの処理により、図19及び図20で説明したような判定結果R1tがレジスタR2に保持される。そして、レジスタR2の値が全ビット“0”か否かを判定し(S76)、そうでなければエラーが有るので、訂正可能な1ビットエラーに対してはサブルーチンとして後述のエラー訂正処理を実行する。そして、レジスタFMPDR0が示す領域に、レジスタR4の下

位 1 6 ビットの値をストアし (S 7)、レジスタ F P F R にパス情報をセットする。尚、特に図示はしないがステップ S 7 6 ではエラーが訂正不能であるか否かも判定し、訂正不能であれば図 2 2 で説明した訂正不能エラー発生の通知処理を行って例外処理を待ってもよい。

【 0 0 9 2 】

《エラー訂正処理の具体例》

図 2 8 には 1 ビットの誤りに対するエラー訂正処理の詳細が示される。エラー訂正処理プログラムにおける C P U 2 の汎用レジスタ及びフラッシュコントロールモジュール 6 内のレジスタの利用形態はエラー判定処理の場合と同じである。図 2 7 において、先ず、レジスタ R 5 に E C C T L B の先頭検索アドレス X をセットし、レジスタ R 6 に E C C T L B の検索ストップアドレスをセットする (S 8 0)。レジスタ R 7 にビット反転用マスクデータ h' 0 0 0 1 をセットする (S 8 1)。そして R 5 の値が R 6 の値になるまで以下の処理を繰り返す。即ち、レジスタ R 5 の検索アドレスに対応する生成用データを E C C T L B からレジスタ R 9 にストアする。そして、R 9 の値が前記レジスタ R 2 の判定結果 R 1 t に一致するかが判別され、一致していれば、その時の R 7 のビット反転用マスクデータにおける “1” のビット位置に対応するユーザデータのビット位置に誤りがある。前記エラー判定処理のステップ S 7 7 で説明したようにレジスタ R 4 の下位 1 6 ビットを切出して、正規のユーザデータとするから、そのとき、レジスタ R 7 のマスクデータとレジスタ R 4 のデータとを対応ビット毎に排他的論理和を採ることによって、マスクデータのビット “1” に対応するユーザデータのビットだけが論理値反転され、エラー訂正されたユーザデータがレジスタ R 4 に返される (S 8 5)。ステップ S 8 4 において R 9 の値が R 2 の値に一致しなければ、レジスタ R 5 に h' 0 1 を加算して、E C C T L B の検索アドレスを次アドレスに進め (S 8 6)、レジスタ R 7 のマスクデータを 1 ビットシフトし (S 8 7)、再度ステップ S 8 3 の処理に戻る。ステップ S 8 3 ~ S 8 7 の処理は E C C T L B の検索アドレスがストップアドレスになるまで繰り返されて (S 8 2)、当該サブルーチンを終了する。このサブルーチンが終了されたとき、レジスタ R 4 には誤り訂正された正規のユーザデータを含んでいる。

【0093】

《マルチチップのデータ処理システム》

図29にはマルチチップのデータ処理システムが例示される。同図に示されるデータ処理システムはデータプロセッサ50とフラッシュメモリ51が夫々別々に半導体集積回路化されてバス52で接続され、その他に、単一又は複数の半導体集積回路で構成された周辺回路53がバス52に接続される。データプロセッサ50はCPU65、RAM66、ROM67、及びI/O68を有する。フラッシュメモリ51は汎用フラッシュメモリであり、フラッシュメモリセルがマトリクス配置されたメモリセルアレイ70、XDE・DRV71、YDE72、TGN73、VGN74、YSW75、SAA76、書き込み・消去制御回路77を有する。フラッシュメモリ51の基本的な構成は前記フラッシュメモリ5と同様であるからその詳細な説明は省略するが、書き込み及び消去を制御するロジック回路として書き込み・消去制御回路77を専用に備える。フラッシュメモリ51の動作はCPU65から与えられるコマンド並びにアクセス制御信号によって決定される。CPU65はフラッシュメモリ51の位置部の記憶領域70Eを頻繁に書き換えるパラメータデータのようなデータの記憶領域として用いる。前述と同様にその領域70Eは他の領域よりも書き換え保証回数を向上させる為にECC処理の対象とされる。ROM67は前記領域70Eをアクセスしてデータ処理を行うユーザプログラム67Pを保有する。このユーザプログラム67Pは、前記領域70Eにパラメータデータを書き込むときに実行されるECCコード生成プログラム67P1、前記領域70Eから読み出したECCコード付加データに対するエラー判定及び誤り訂正を行う為のエラー訂正プログラム67P2を有する。前記ECCコード生成プログラム及びエラー訂正プログラムは所定のユーザプログラムから呼び出されて実行される。

【0094】

上記マルチチップのデータ処理システムにおいても図1のシングルチップのマイクロコンピュータ化されたデータ処理システムと同様に、ECCコードによる記憶領域の利用の無駄を省いて記憶情報の信頼性を向上させることができ、ECCコードによる記憶領域の利用の無駄を省いて記憶情報の書換え保証回数を向上

させることが可能であり、デバイスの特性に合った E C C 方式を選択してデータに対する E C C コードのオーバーヘッドを小さくし記憶領域の利用効率を最大限とすることが可能である等の効果を得ることができる。

【 0 0 9 5 】

《多値フラッシュメモリへの考慮》

前記フラッシュメモリ 5, 5 1 は、1 個のフラッシュメモリセルに 2 ビット以上の記憶情報を保持させることが可能な多値フラッシュメモリであってもよい。すなわち、1 個のフラッシュメモリセルは、情報記憶に際して複数ビットの書き込みデータで指定される 4 種類以上の閾値電圧の中の一つの閾値電圧に設定され、情報読み出しに際して閾値電圧の状態を対応する複数ビットの記憶情報として出力する、1 個のフラッシュメモリセルの記憶情報を複数ビット化したメモリである。ここでは、一つのフラッシュメモリセルに 2 ビットの情報を書き込むことができ、かつその情報を読み出すことができるフラッシュメモリを一例とする。このようなフラッシュメモリが実現しようとする多値情報記憶技術において、一つのメモリセルの情報記憶状態は、例えば消去状態（“1 1”）、第 1 の書き込み状態（“1 0”）、第 2 の書き込み状態（“0 0”）、第 3 の書き込み状態（“0 1”）の中から選ばれた一つの状態とされる。全部で 4 通りの情報記憶状態は、2 ビットのデータによって決定される状態とされる。即ち、2 ビットのデータを一つのメモリセルで記憶する。この 4 値のデータと閾値電圧との関係は、図 3 0 の閾値電圧分布図に示される通りである。フラッシュメモリセルの記憶データの値と閾値電圧との関係を図 3 0 のように規定すると、情報記憶後に、閾値電圧が不所望に変化しても隣りの閾値電圧のデータとは相互に 1 ビットしか相違しないようになる。したがって、データエラーを生じても殆どが 1 ビットエラーになり、1 ビットエラーに対して訂正可能な E C C 処理によるデータの信頼性を高く維持することが可能になる。換言すれば、多値フラッシュメモリセルの閾値電圧が近いところをデータハミング距離が 1 になるようにすれば、相対的に少ないビット数の E C C コードで高い信頼性を得ることができ、書換え保証回数の向上が容易になる。

【 0 0 9 6 】

以上本発明者によってなされた発明を実施形態に基づいて具体的に説明したが、本発明はそれに限定されるものではなく、その要旨を逸脱しない範囲において種々変更可能であることは言うまでもない。

【 0 0 9 7 】

例えば、フラッシュメモリセルはフローティングゲートとコントロールゲートの縦積み構造に限定されず、MOSトランジスタのゲート電極をフローティングゲート電極とし当該ゲート電極を延在させて形成したMOSゲート容量を介してチャネル領域をコントロールゲートに用いるようなデバイス構造などを採用してもよい。また、不揮発性記憶素子はフラッシュメモリに限定されず、MNOS（メタル・ナイトライド・オキサイド・セミコンダクタ）トランジスタを記憶素子とするEEPROM（エレクトリカリ・イレーザブル・アンド・プログラマブル・リード・オンリ・メモリ）のような不揮発性メモリ、或いは誘電体メモリ等であってもよい。

【 0 0 9 8 】

データ処理システムは、1個の半導体チップに前記不揮発性メモリ及び中央処理装置が形成されたシングルチップのマイクロコンピュータとして実現することが可能であり、その一方において、前述の如く前記データ処理システムは、前記不揮発性メモリ及び中央処理装置が夫々別々の半導体チップに形成されたマルチチップ形態で実現してもよい。そして、データ処理システムは、シングルチップ及びマルチチップのマイクロコンピュータに限定されず、フラッシュメモリを内蔵したグラフィックスコントローラ、DRAMを専用ロジック回路と共に混載したシステムLSI、その他のマルチチップによる電子回路に広く適用する事ができる。

【 0 0 9 9 】

【発明の効果】

本願において開示される発明のうち代表的なものによって得られる効果を簡単に説明すれば下記の通りである。

【 0 1 0 0 】

すなわち、一部の記憶領域にだけECCコードの付加や誤り訂正を行なって書

換え保証回数を向上させるから、記憶領域に拘わらず全てのライトデータに対して区別なくECCコードを付加する構成に比べて、実質的に無用のECCコードによる記憶領域の無駄を省くことができる。更に、ECC処理をソフトウェアで対処するから不揮発性メモリのデバイス特性に合ったECC訂正能力を容易に選択することができる。このように、データ処理システムにおいてECCコードによる記憶領域の利用の無駄を省いて不揮発性メモリに記憶された情報の信頼性を向上させることができ、また、ECCコードによる記憶領域の利用の無駄を省いて不揮発性メモリにおける記憶情報の書換え保証回数を向上させることができる。更に、デバイスの特性に合ったECC方式を選択でき、データに対するECCコードのオーバーヘッドを小さくして記憶領域の利用効率を最大限とすることが可能である。

【0101】

ECCコードが付加されたデータに対する誤り判定及び訂正の処理によるデータリード動作の遅延を抑えることが可能である。

【0102】

積和演算などを行うことなくECCコードの生成を効率的に行なうことができ、ECCコードが付加されたデータの誤り判定を効率的に行なうことができる。

【図面の簡単な説明】

【図1】

本発明の一例に係るシングルチップのマイクロコンピュータのブロック図である。

【図2】

CPUの具体例を示すブロック図である。

【図3】

書換え保証回数を向上させる記憶領域、ECCコード生成プログラム及びエラー訂正プログラムの記憶領域のマッピング例を示す説明図である。

【図4】

ユーザデータのビット数に対するECCコードのビット数による誤り訂正能力とオーバーヘッドとの関係を例示する説明図である。

【図 5】

ユーザデータと E C C コードを一つの配列データ中に対応付けての配列するデータフォーマットと、ユーザデータと E C C コードとを別の配列データとして対応付けるデータフォーマットとを例示する説明図である。

【図 6】

ユーザデータと E C C コードとを一つの配列データ中に対応付けて書き込むときの処理手順を例示する説明図である。

【図 7】

ユーザデータと E C C コードとを別の配列データとして対応付けて書き込むときの処理手順を例示する説明図である。

【図 8】

図 6 で説明したレコード配列を有する E C C コード付きデータをリードするときの処理手順を例示する説明図である。

【図 9】

図 7 で説明したユーザデータと E C C コードが別々の配列データとされる E C C コード付きデータをリードするときの処理手順を例示する説明図である。

【図 1 0】

E C C コード生成プログラム及びエラー訂正プログラムの実行速度を向上させる為の一例を示す説明図である。

【図 1 1】

エラー判定によるオーバヘッドを見掛け上解消するための一例を示す説明図である。

【図 1 2】

エラー判定によるオーバヘッドを見掛け上解消するための別の一例を示す説明図である。

【図 1 3】

1 6 ビットデータに対する 1 ビット訂正のための行列テーブル 4 0 の一例を示す説明図である。

【図 1 4】

ECCコードを付加した符号語を生成する原理的手法の説明図である。

【図 1 5】

H' 8 0 4 1 の 1 6 ビットデータに対し排他的論理和を用いて ECC コードを生成する具体例の説明図である。

【図 1 6】

排他的論理和演算のための演算手段を ECC コード 1 ビット分の構成を代表として例示するブロック図である。

【図 1 7】

図 1 5 の例において排他的論理和を採った具体的な演算結果である ECC コードを例示する説明図である。

【図 1 8】

エラー判定処理の原理的手法を示す説明図である。

【図 1 9】

図 1 8 の例において前記排他的論理和による誤り無の判定結果が例示される説明図である。

【図 2 0】

図 1 9 に対して 1 ビット誤りの有る判定結果が得られる場合を例示する説明図である。

【図 2 1】

2 ビット以上の訂正不能な誤りが発生している時の判定結果を列挙する説明図である。

【図 2 2】

訂正不能エラーに対して例外処理を実行可能にする例を示す説明図である。

【図 2 3】

ユーザデータと ECC コードの符号語に採用される固定フォーマットの一例を示す説明図である。

【図 2 4】

図 2 3 の固定フォーマットの更に詳細を例示する説明図である。

【図 2 5】

行列テーブルのデータを保有する E C C テーブルを例示する説明図である。

【図 2 6】

E C C コード生成プログラムによる処理手順を例示するフローチャートである。

【図 2 7】

図 2 6 の手順で生成された E C C コード付加データをリードしたときのエラー判定処理の詳細を示すフローチャートである。

【図 2 8】

1 ビットの誤りに対するエラー訂正処理の詳細を示すフローチャートである。

【図 2 9】

マルチチップのデータ処理システムを例示するブロック図である。

【図 3 0】

多値フラッシュメモリにおける 4 値のデータと閾値電圧との関係を例示する説明図である。

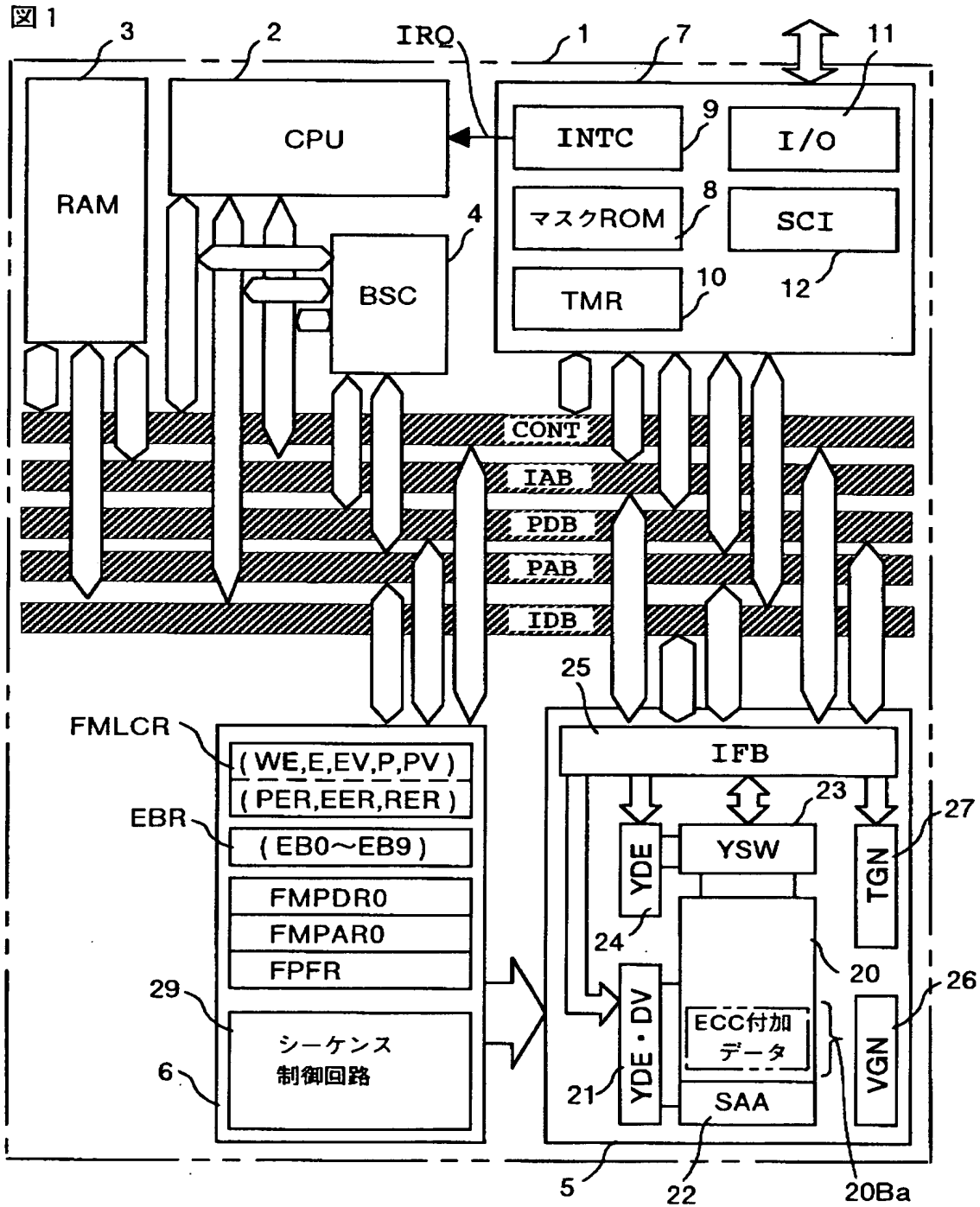
【符号の説明】

- 1 マイクロコンピュータ
- 2 C P U
- 3 R A M
- 5 フラッシュメモリ
- 6 フラッシュコントロールモジュール
- 8 マスク R O M
- 2 0 メモリセルアレイ
- 2 0 A ブート領域
- 2 0 B ユーザ領域
- 2 0 B a 相対的に書換え保証回数の高い記憶領域
- 2 0 B b 相対的に書換え保証回数の低い記憶領域
- 2 1 E C C コード生成プログラム
- 2 2 エラー訂正プログラム
- D A 1 ユーザデータの配列

DA2 固定フォーマットの符号語の配列
DA3 ECCコードのみから成るデータ配列
DATA1 ECCコード付加パラメータデータ
DATA2 エラー判定・訂正済パラメータデータ
RER リードエラービット
30 リードエラーフラグ
40 行列テーブル
41 生成行列
42 検査行列
M データ (ユーザデータ)
41A 生成行列の内ECCに関する部分を抜き出した行列
P ECCコード
C 符号語
Rlt エラー判定の演算結果
Cer 1ビットの誤りを含む符号語
Ber 誤りビット
50 データプロセッサ
51 フラッシュメモリ
65 CPU
66 RAM
67 ROM
70 メモリセルアレイ
70E 相対的に書換え保証回数の高い記憶領域

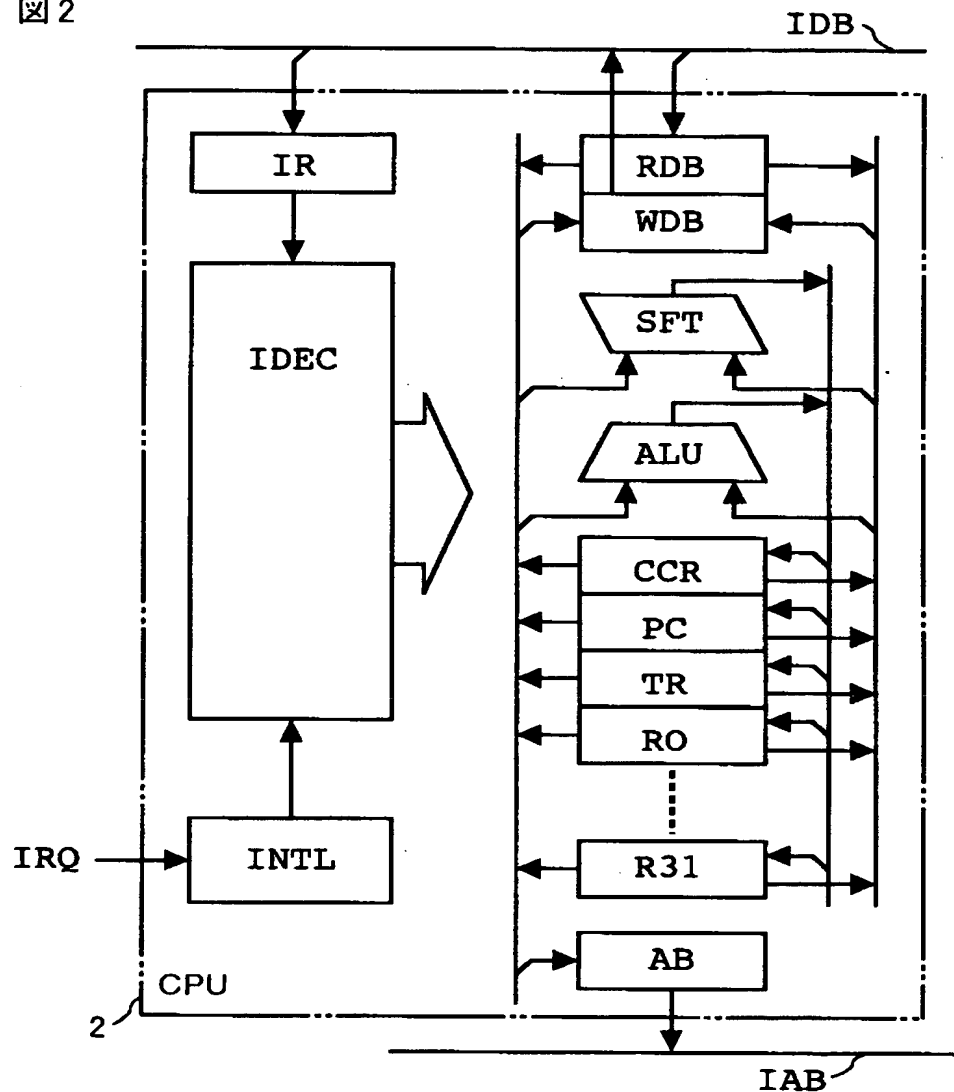
【書類名】 図面

【図 1】



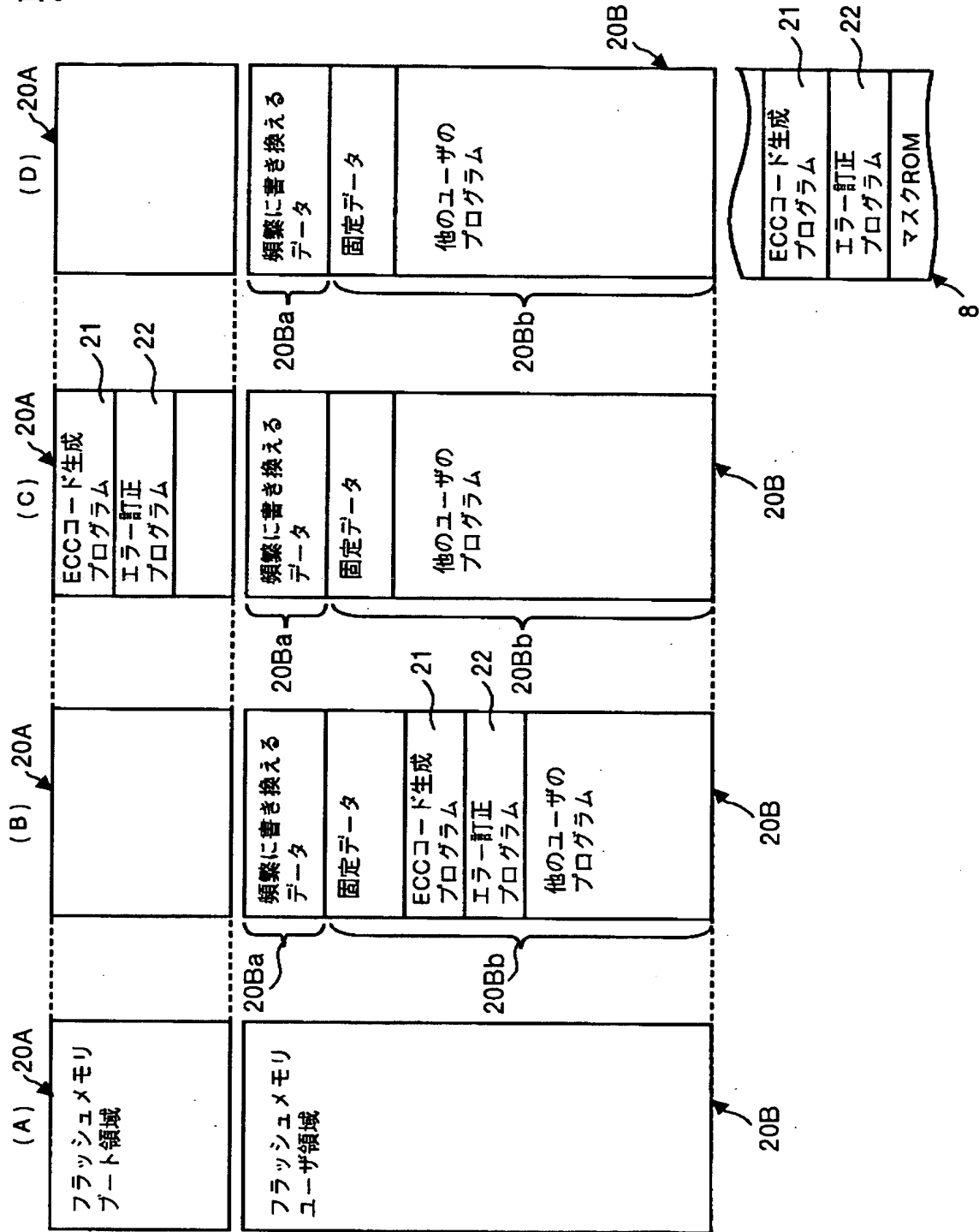
【図 2】

図 2



【図 3】

図 3



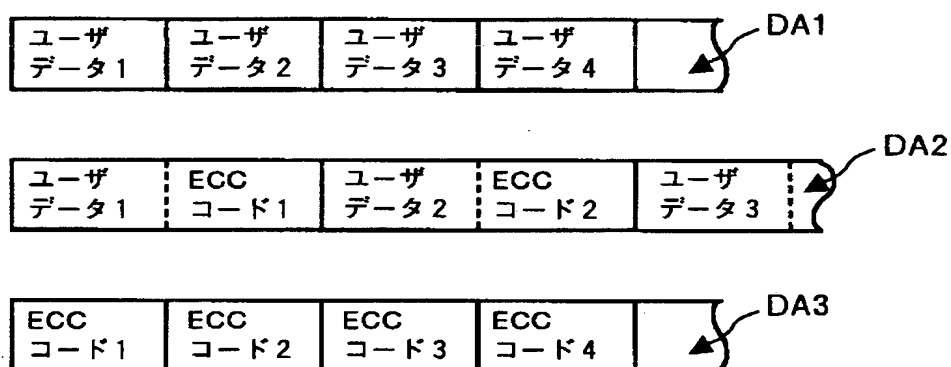
【図 4】

図 4

ユーザデータビット数	ECCコードビット数	オーバーヘッド	訂正能力
8ビット	4ビット	50%	8ビット中1ビット
16ビット	5ビット	31%	16ビット中1ビット
32ビット	6ビット	19%	32ビット中1ビット
64ビット	7ビット	11%	64ビット中1ビット
128ビット	8ビット	6%	128ビット中1ビット
256ビット	9ビット	4%	256ビット中1ビット

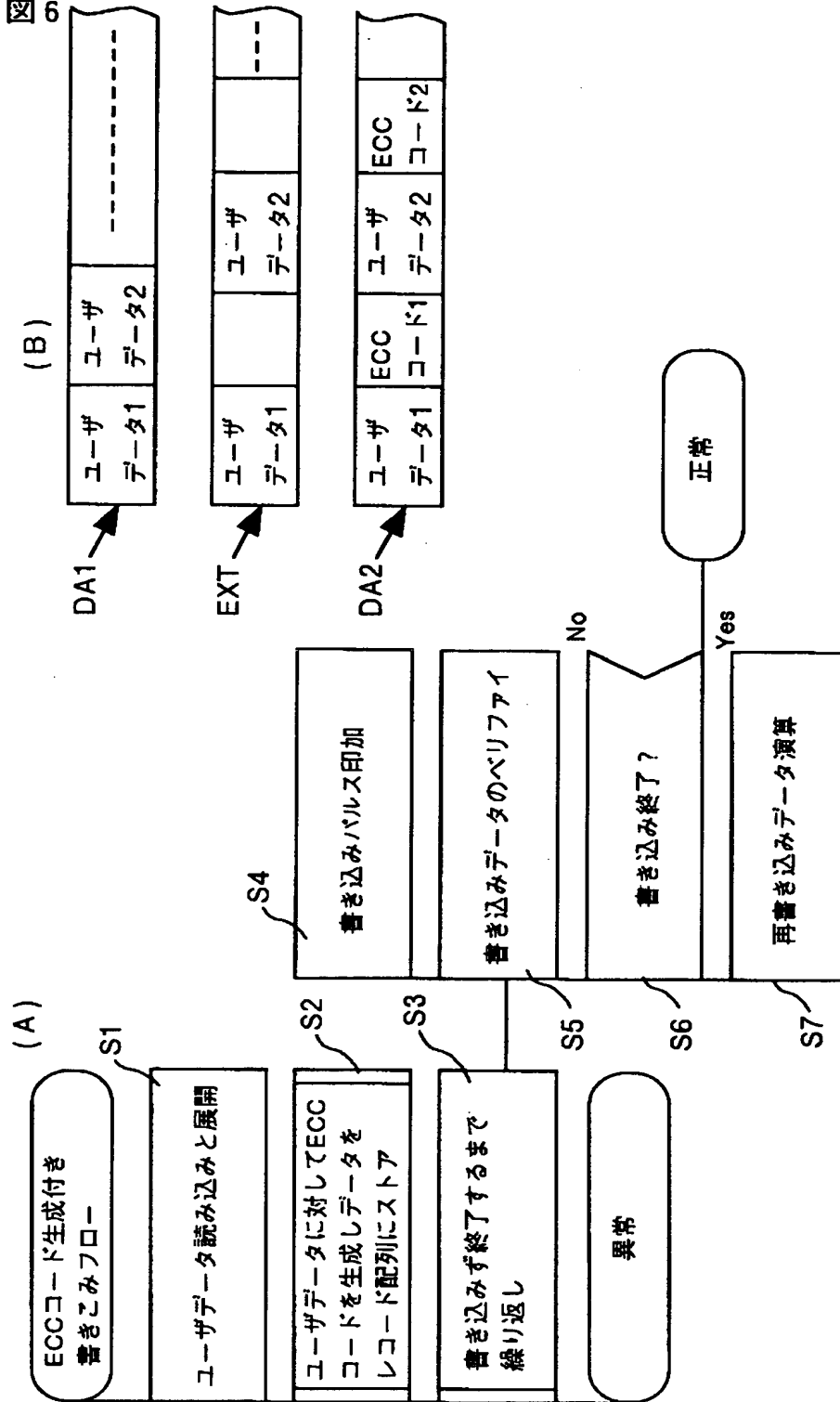
【図 5】

図 5



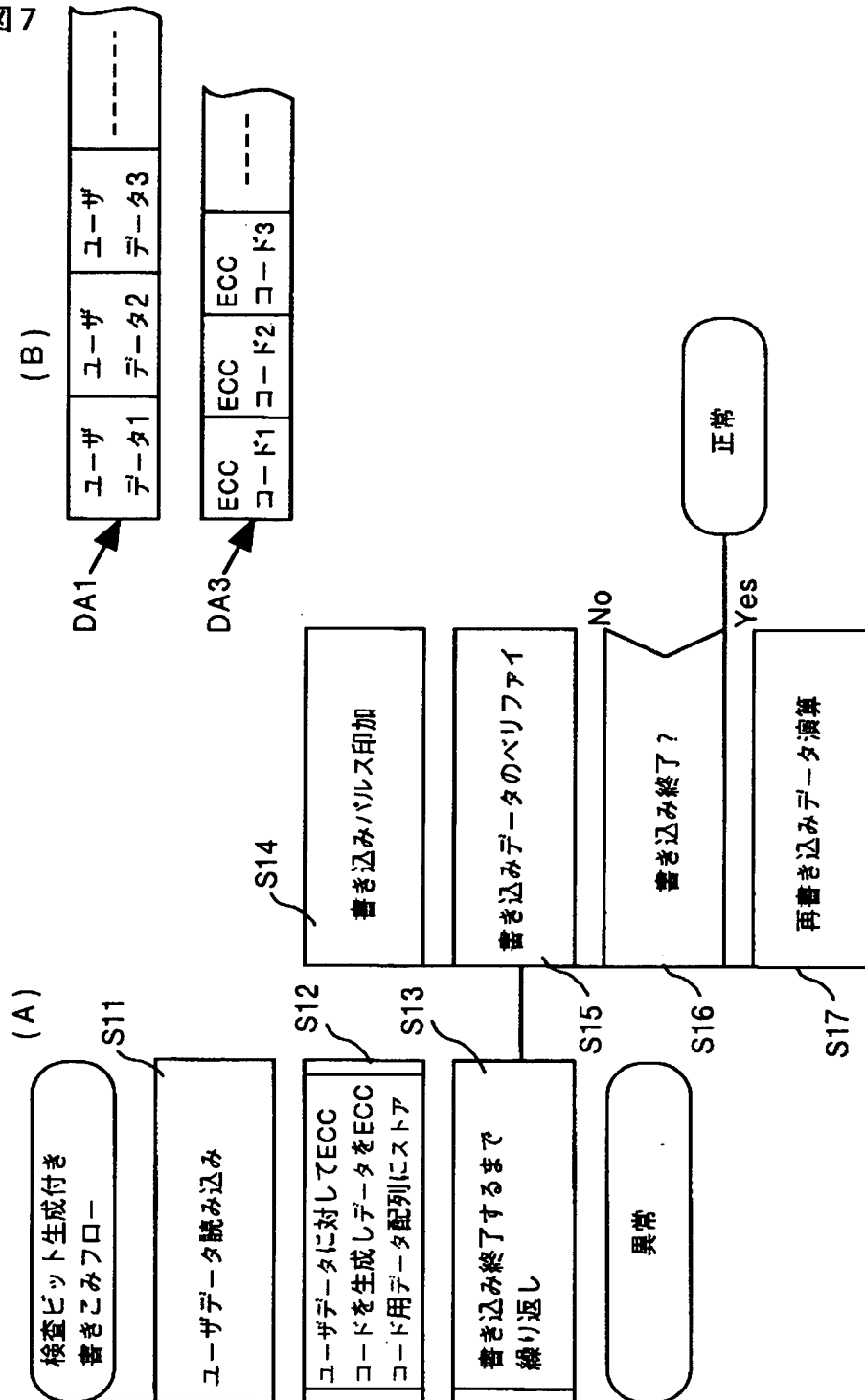
【図 6】

図 6



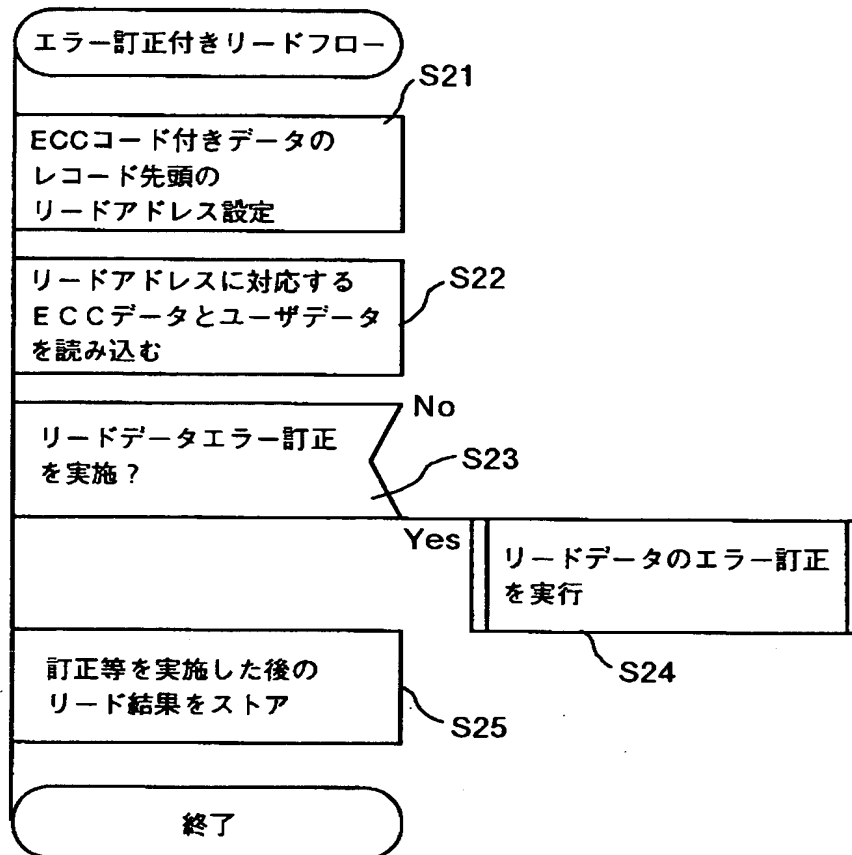
【図 7】

図 7



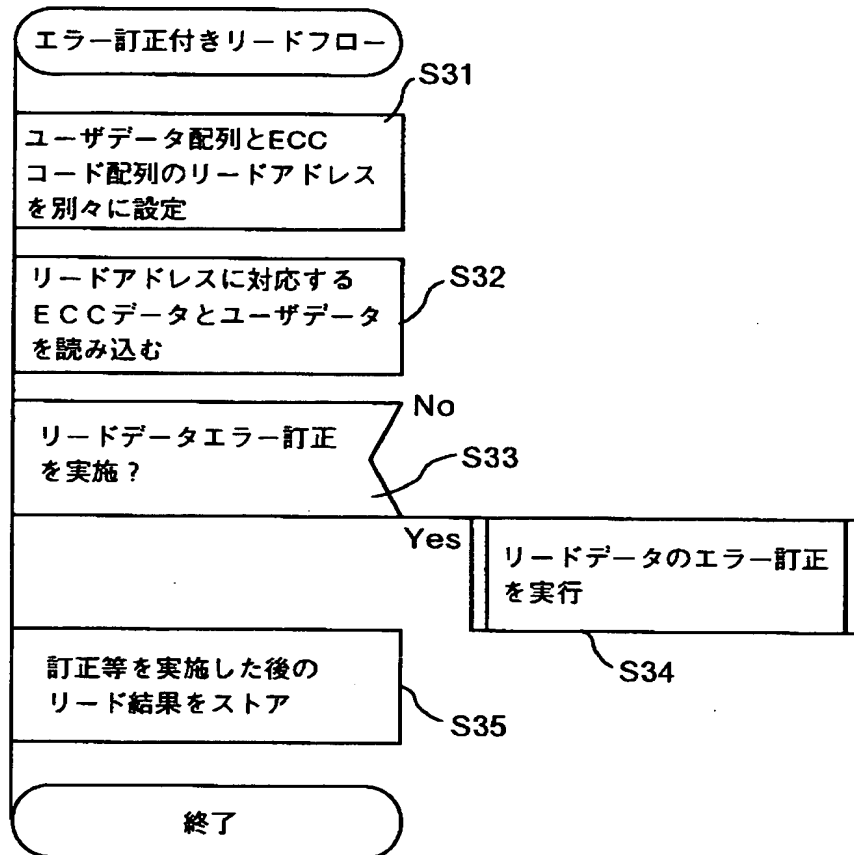
【図 8】

図 8



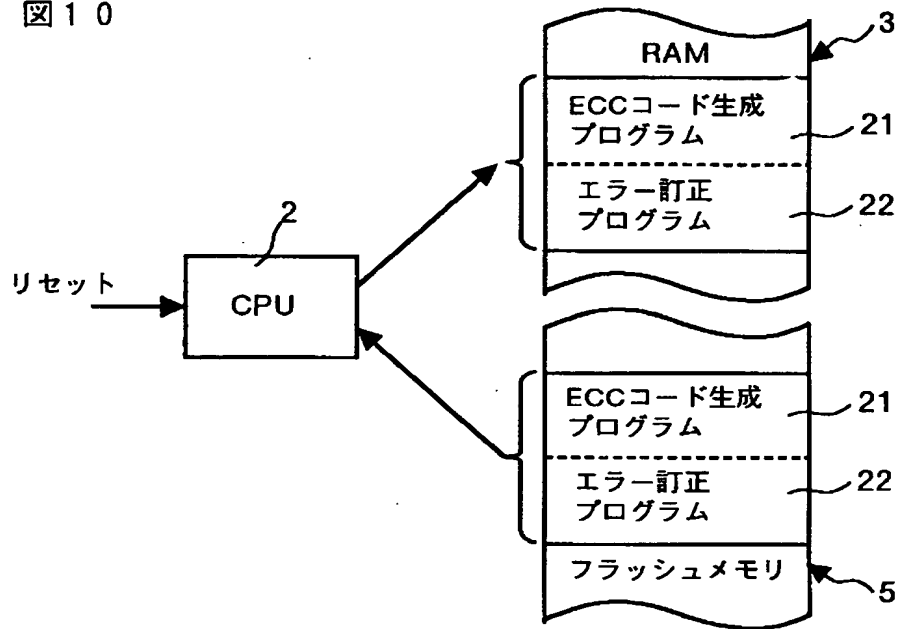
【図 9】

図 9



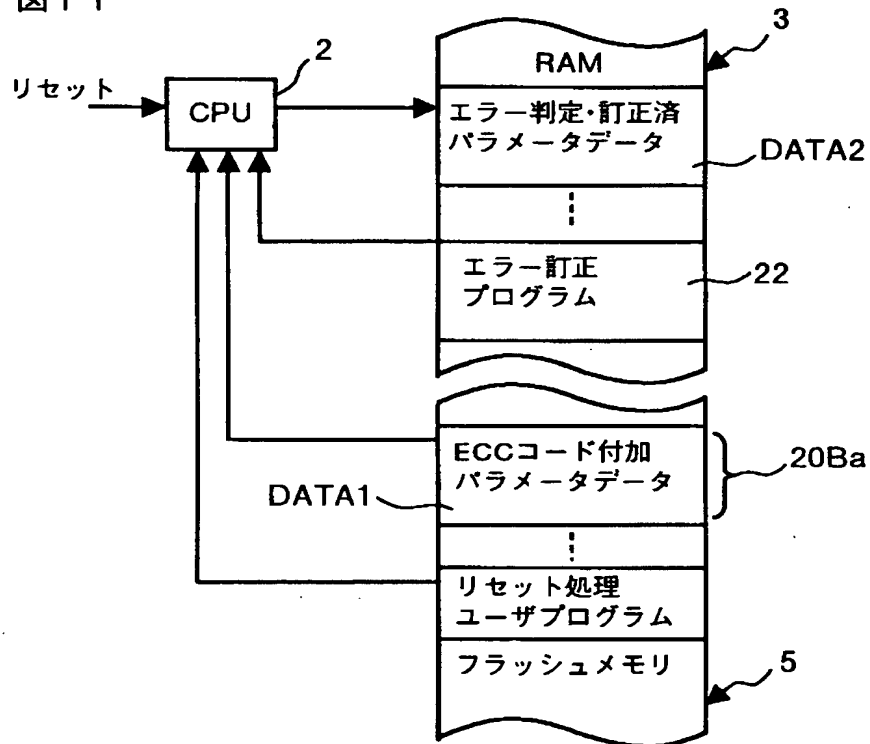
【図10】

図10

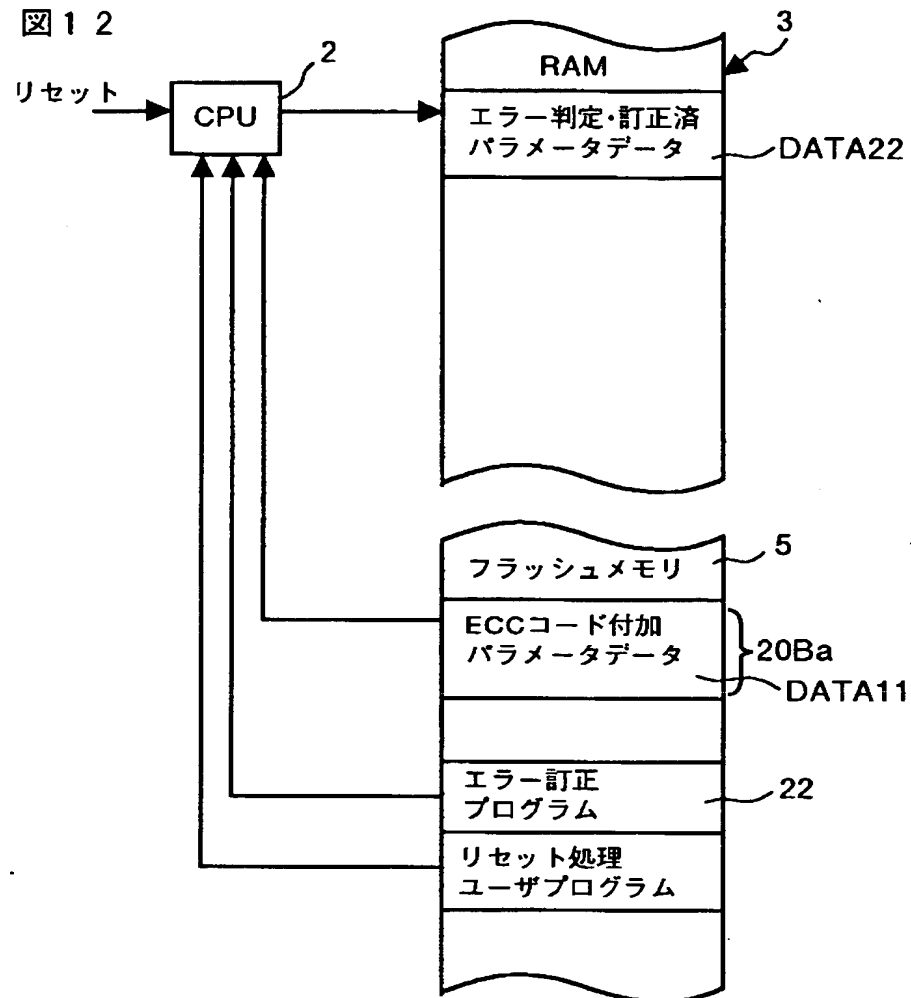


【図11】

図11

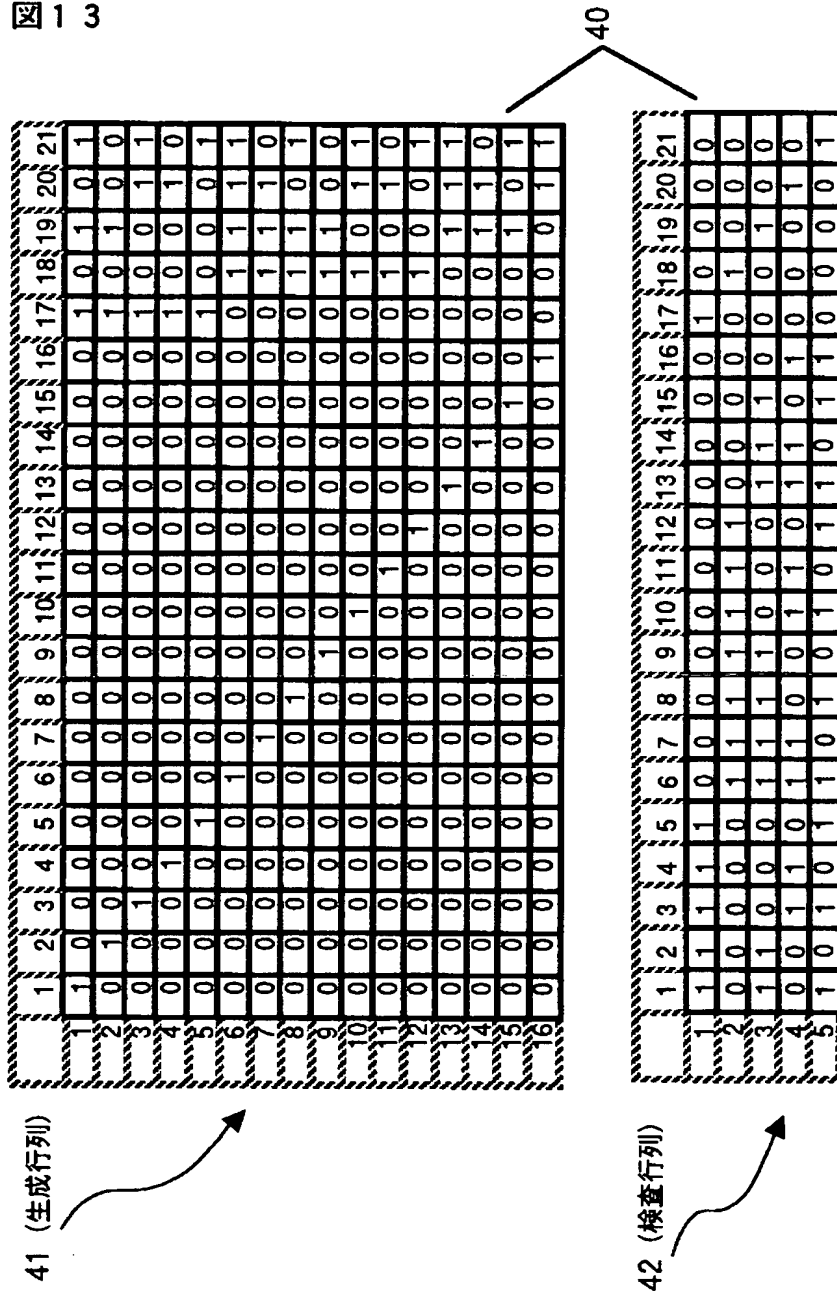


【図 1 2】



【図 1 3】

図 1 3



【図 1 4】

図 1 4

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
m15	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1
m14	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
m13	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1
m12	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
m11	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
m10	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
m9	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	0
m8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	1
m7	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0
m6	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	1
m5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0
m4	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1
m3	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1
m2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0
m1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1
m0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1

41

M

【図15】

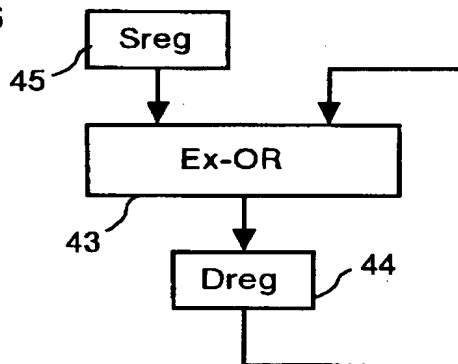
図15

		1	2	3	4	5	
1	1	0	1	0	1	1	1
0	1	0	1	0	0	0	2
0	1	0	0	1	1	1	3
0	1	0	0	1	0	0	4
0	1	0	0	0	1	1	5
0	0	1	1	1	1	1	6
0	0	1	1	1	1	0	7
0	0	1	1	0	1	1	8
0	0	1	1	0	0	0	9
1	0	1	0	1	1	1	10
0	0	1	0	1	0	0	11
0	0	1	0	0	1	1	12
0	0	0	1	1	1	1	13
0	0	0	1	1	0	0	14
0	0	0	1	0	1	1	15
1	0	0	0	1	1	1	16

M → 41A

【図16】

図16



【図17】

図17

1行目	1	1	0	1	0	1
10行目	1	0	1	0	1	1
16行目	1	0	0	0	1	1
Ex-OR結果		1	1	1	0	1

P

【図 1 8】

図 1 8

42

C

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
2	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	1	0	0	0
3	1	1	0	0	0	1	1	1	1	0	0	0	1	1	1	0	0	0	1	0	0
4	0	0	1	1	0	1	1	1	0	0	1	0	1	1	0	1	0	0	0	1	0
5	0	1	0	1	0	1	1	0	1	0	1	0	1	0	1	1	0	0	0	0	1
	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	1	0	1

【図 1 9】

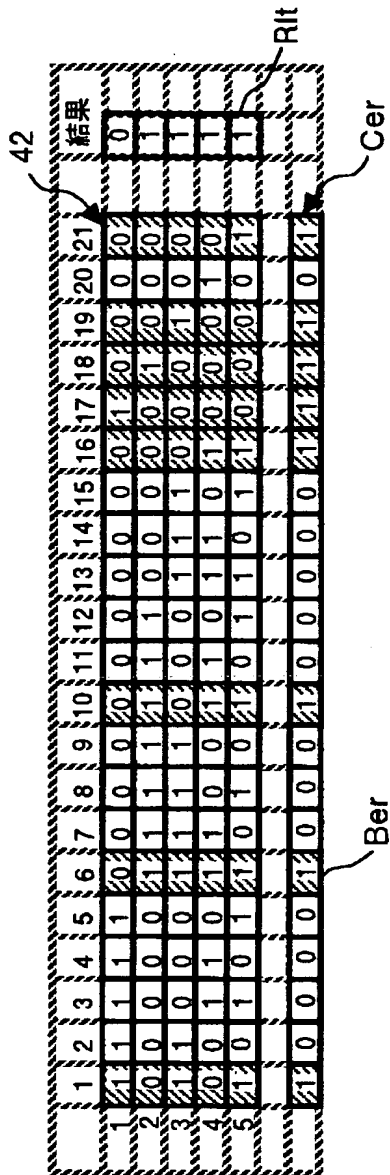
図 1 9

1	10	16	17	18	19	21	結果
1	0	0	1	0	0	0	0
0	1	0	0	1	0	0	0
1	0	0	0	0	1	0	0
0	1	1	0	0	0	0	0
1	1	1	1	1	1	0	0

Rlt

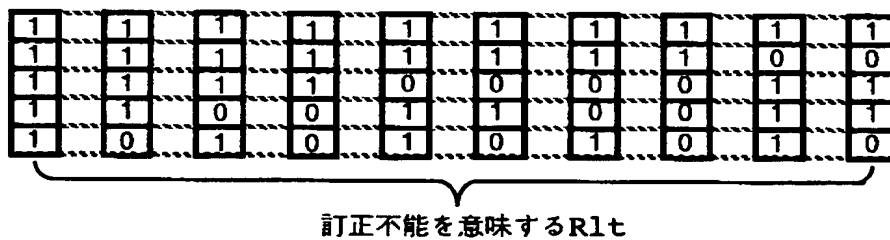
【図 20】

圖 20



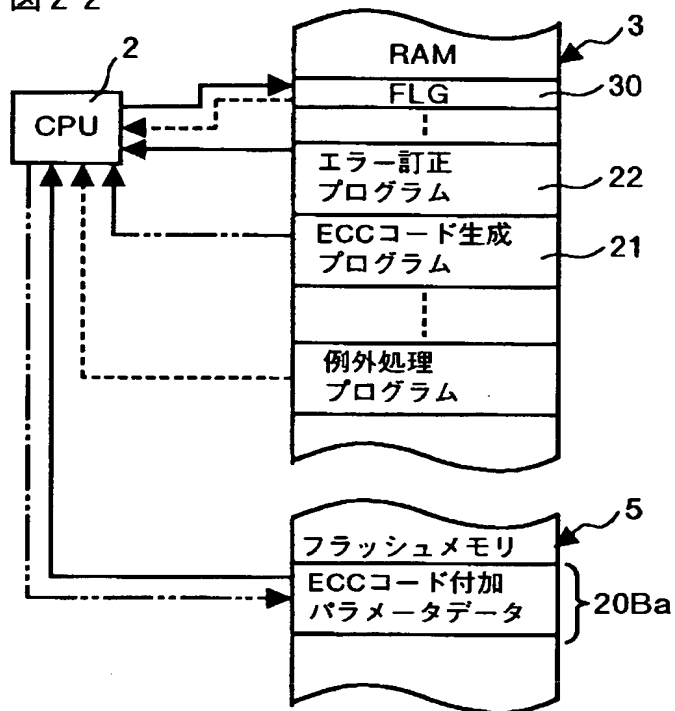
【図 2 1】

圖 21



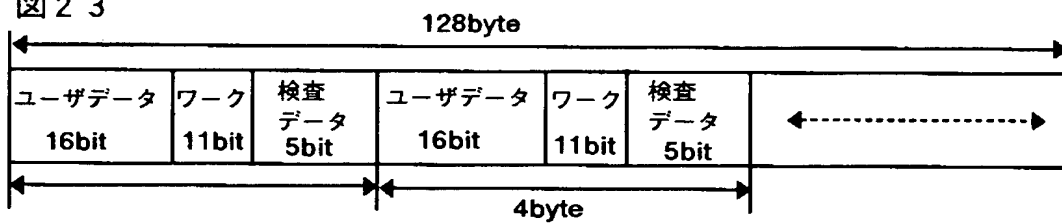
【図 2 2】

図 2 2



【図 2 3】

図 2 3



【図 2 4】

図 2 4

ECCデータブロックの構成

ビット	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
機能名	D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
初期値	ユーザの書き込みデータ															
備考	ユーザの書き込みデータ (エラーコレクションを実施したいデータ)															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	P04	P03	P02	P01	P00
ユーザは、ALL "1" を転送する。															
未使用												符号化検査ビット			

【図 2 5】

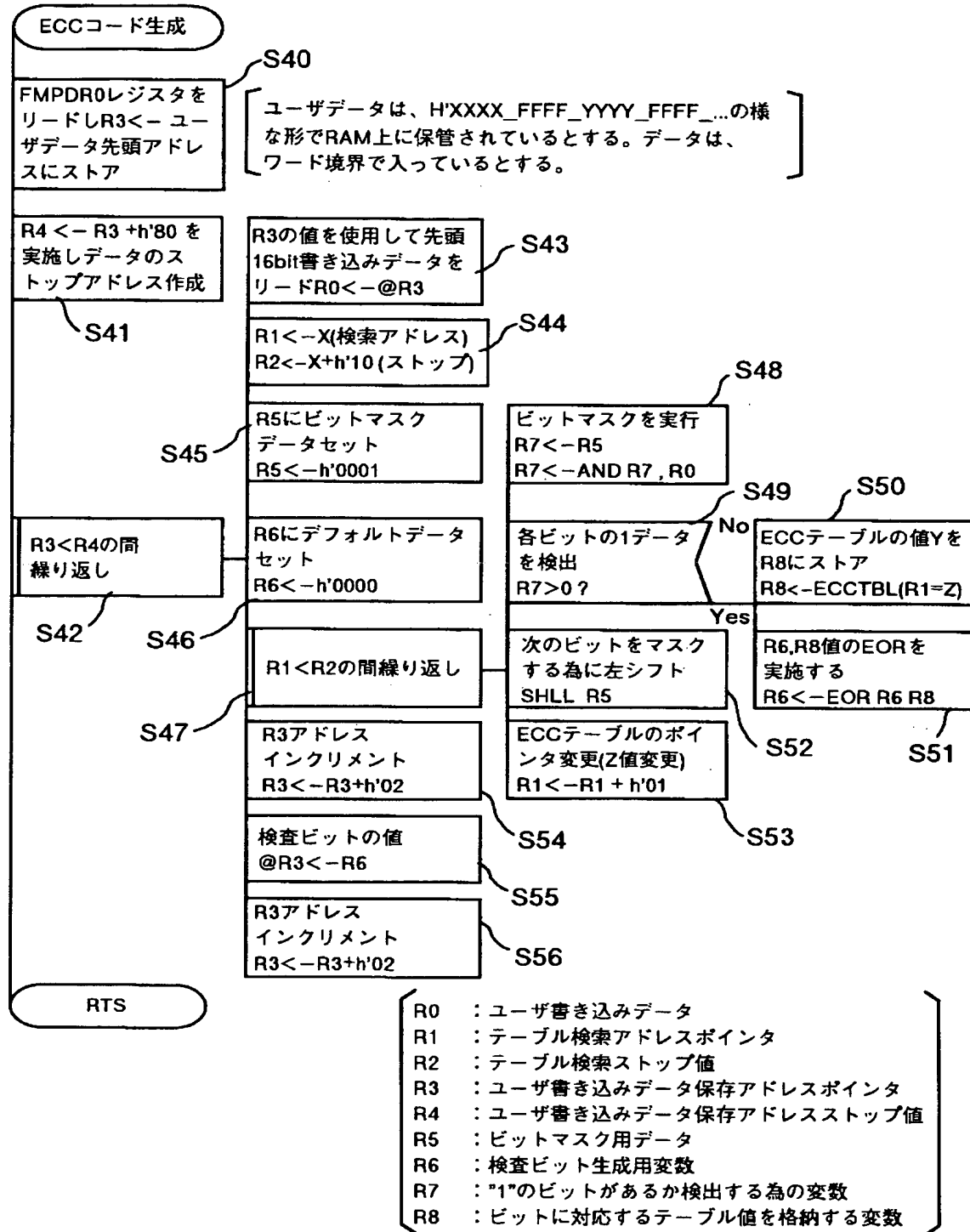
図 2 5

検索アドレス(Z)	生成用データ(Y)	機能名
X	b'0000_0011	D00
X+1	b'0000_0101	D01
X+2	b'0000_0110	D02
X+3	b'0000_0111	D03
X+4	b'0000_1001	D04
X+5	b'0000_1010	D05
X+6	b'0000_1011	D06
X+7	b'0000_1100	D07
X+8	b'0000_1101	D08
X+9	b'0000_1110	D09
X+A	b'0000_1111	D10
X+B	b'0001_0001	D11
X+C	b'0001_0010	D12
X+D	b'0001_0011	D13
X+E	b'0001_0100	D14
X+F	b'0001_0101	D15
X+10	b'0000_0001	P00
X+11	b'0000_0010	P01
X+12	b'0000_0100	P02
X+13	b'0000_1000	P03
X+14	b'0001_0000	P04

ECCテーブル (ECCTLB)

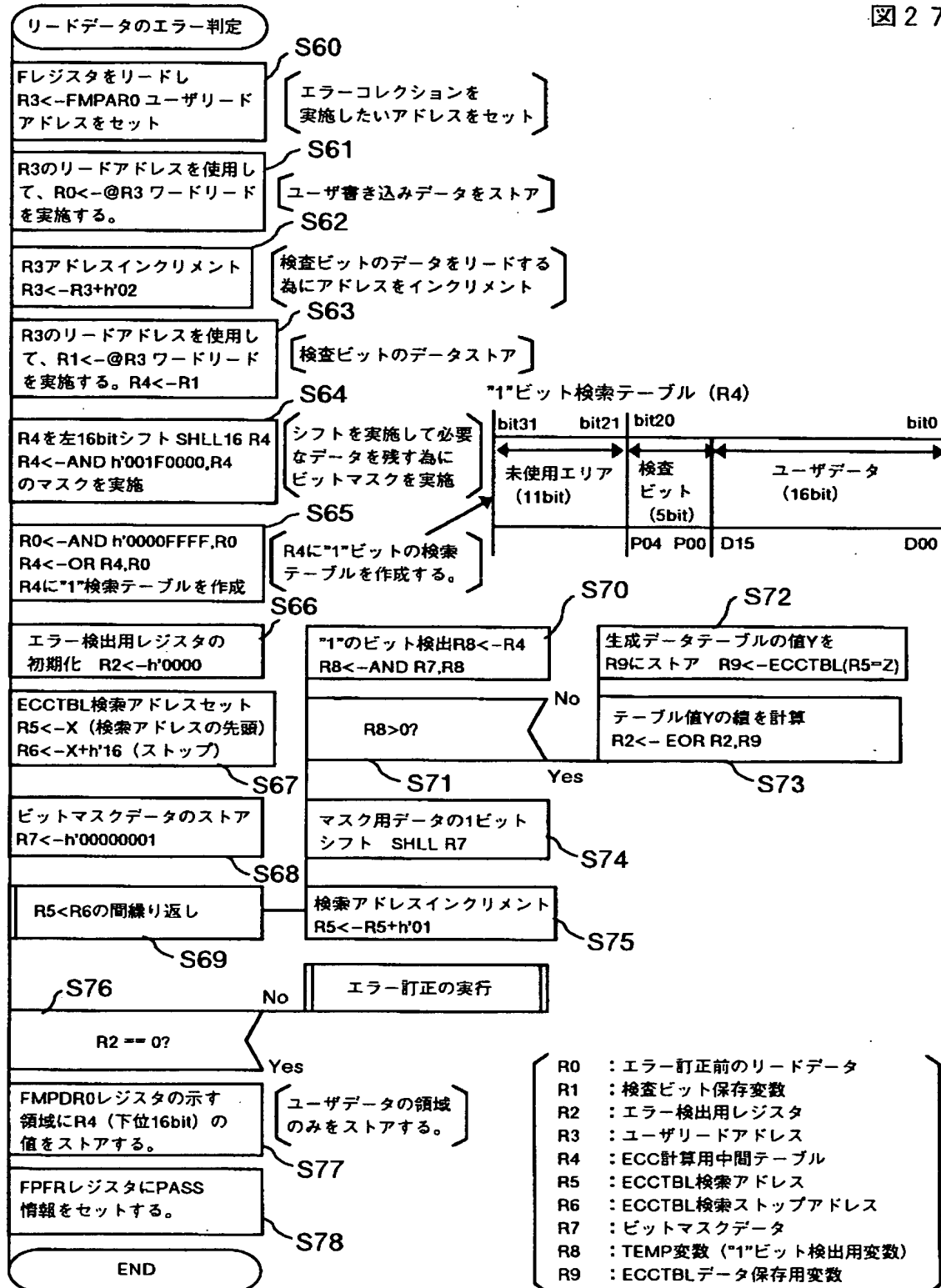
【図 26】

図 26



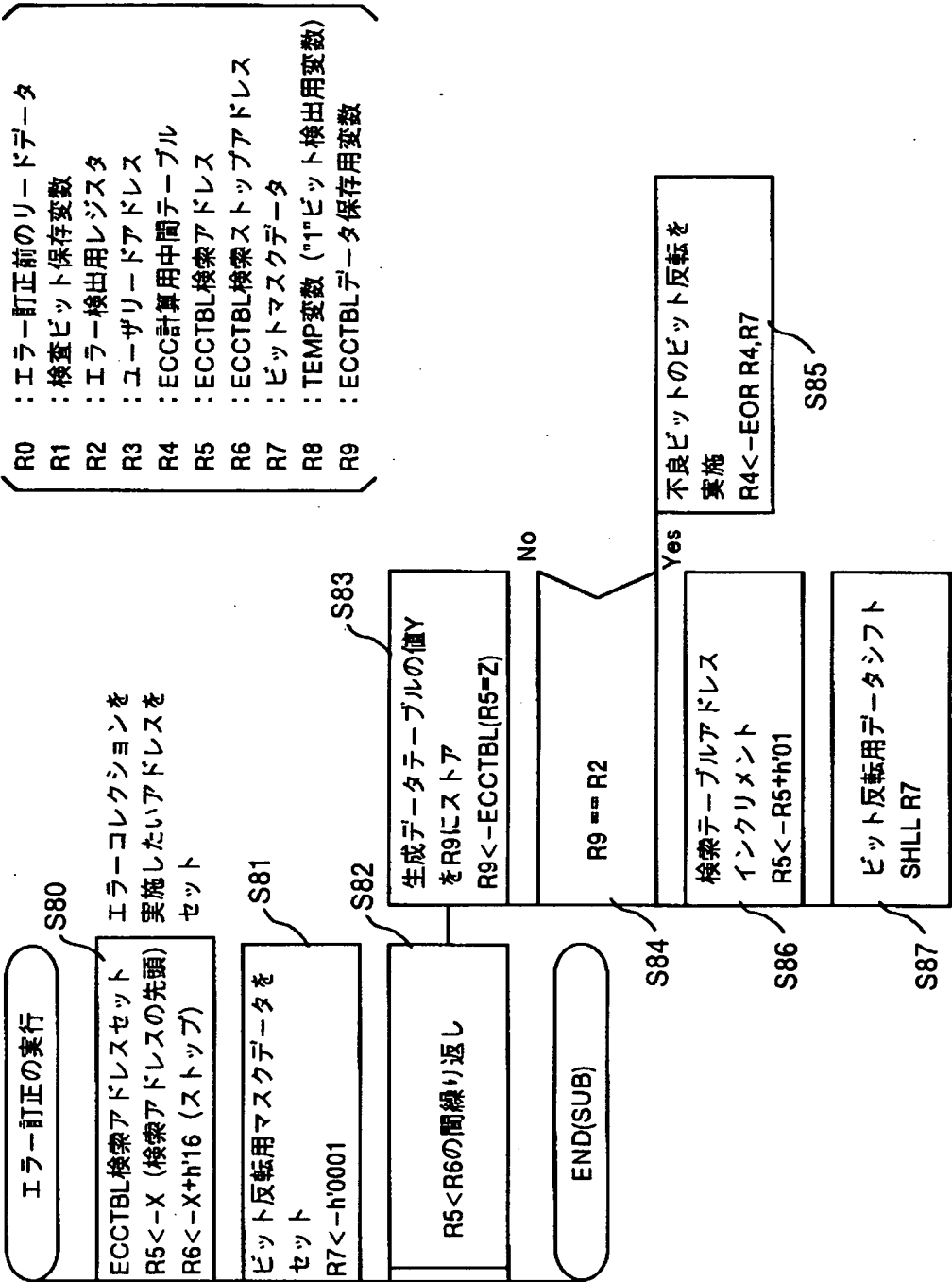
【図 27】

図 27



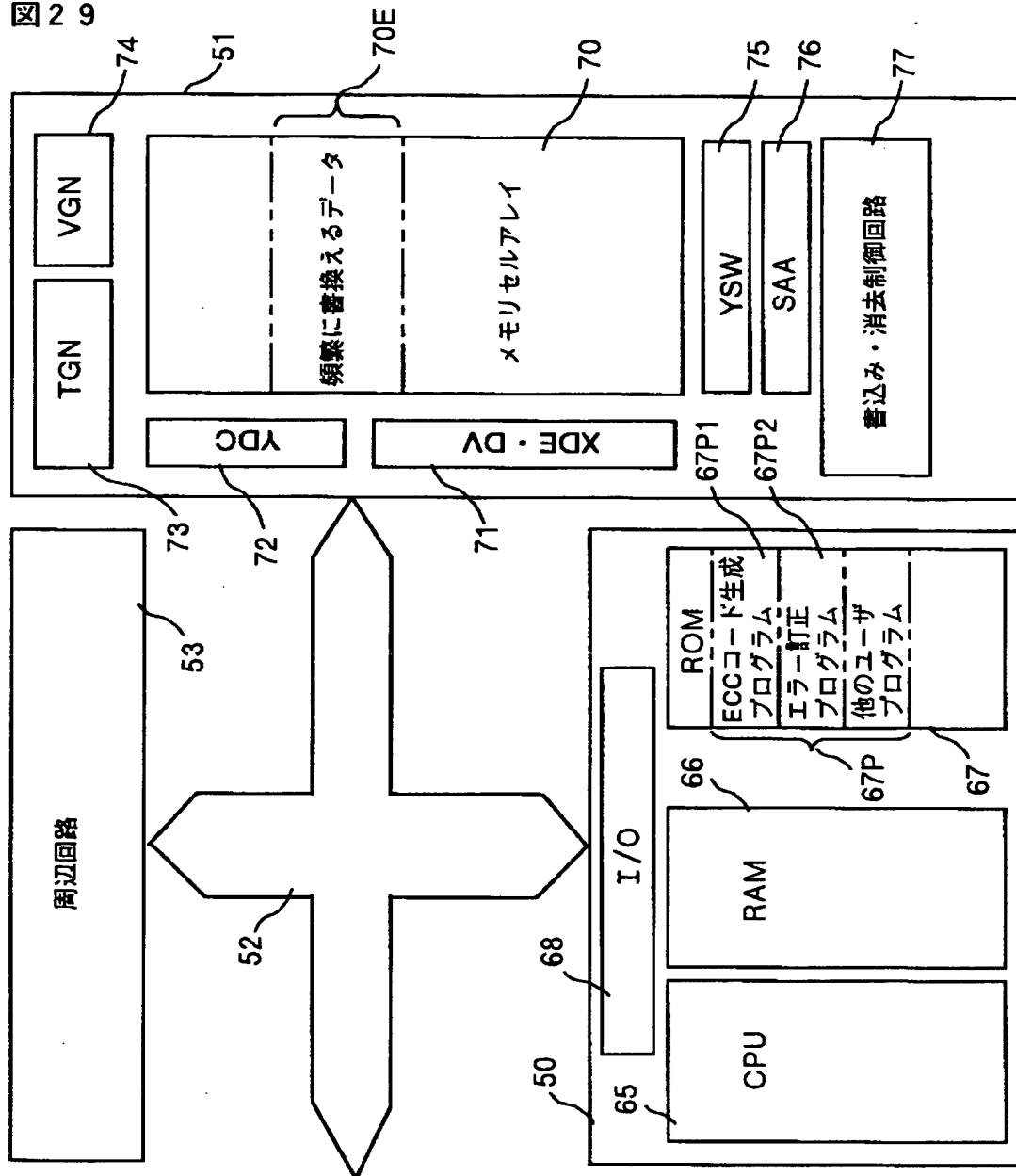
【図 2 8】

図 2 8



【図 29】

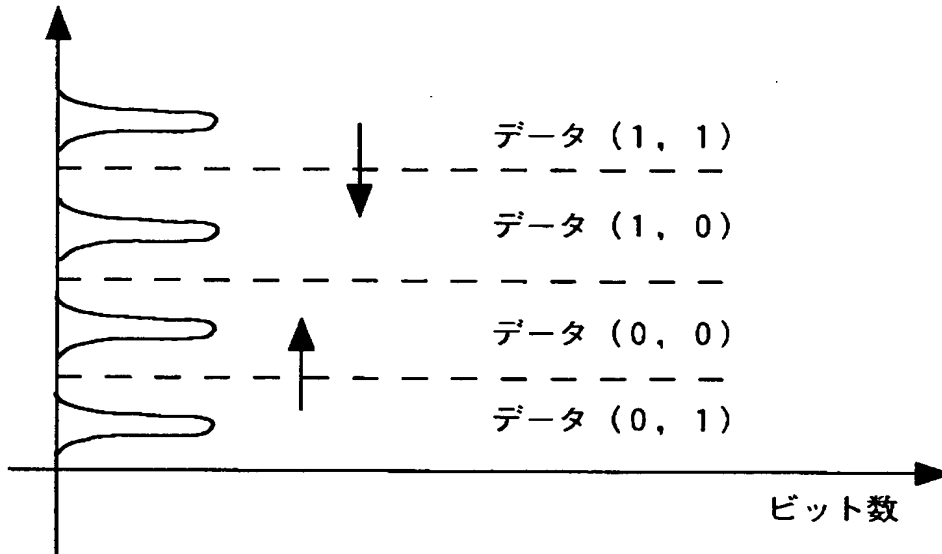
図 29



【図 3 0】

図 3 0

閾値電圧



【書類名】 要約書

【要約】

【課題】 ECCコードによる記憶領域の利用の無駄を省いて不揮発性メモリにおける記憶情報の書換え保証回数を向上させる。

【解決手段】 データ処理システム（１）は、書き換え可能な不揮発性メモリ（５）と、中央処理装置（２）とを有し、前記中央処理装置は、前記不揮発性メモリの指定された一部の記憶領域（２０Ｂａ）に対してだけソフトウェアECC処理の対象とする。一部の記憶領域にだけECCコードの付加や誤り訂正を行なって書換え保証回数を向上させるから、記憶領域に拘わらず全てのライトデータに対して区別なくECCコードを付加する構成に比べて、実質的に無用のECCコードによる記憶領域の無駄を省くことができ、また、ECC処理をソフトウェアで対処するから不揮発性メモリのデバイス特性に合ったECC訂正能力を容易に選択することができる。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日	1990年 8月31日
[変更理由]	新規登録
住 所	東京都千代田区神田駿河台4丁目6番地
氏 名	株式会社日立製作所

出 願 人 履 歴 情 報

識別番号 [0 0 0 2 3 3 5 9 4]

1. 変更年月日	1 9 9 0 年 8 月 3 1 日
[変更理由]	新規登録
住 所	北海道亀田郡七飯町字中島 1 4 5 番地
氏 名	日立北海セミコンダクタ株式会社